# Developing Cybersecurity Capability Forensic Risk Modelling for the Internet of Things

Bryce Antony
MISDF ($1^{st}$ Class), MBA

A thesis submitted to the graduate faculty of Design and Creative Technologies
Auckland University of Technology
in fulfilment of the requirements for the degree of
Doctor of Philosophy (PhD)

School of Engineering, Computer and Mathematical Sciences

Auckland, New Zealand
2020

# Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a University or other institution of higher learning, except where due acknowledgement is made in the acknowledgements.

# Acknowledgements

First and foremost, I would like to thank my supervisor Dr. Brian Cusack for sharing his wealth of knowledge and experience, along with his continual support and advice. I have learned a great deal from Dr. Cusack. I am grateful for his supervision, which was superlative.

I am thankful for Dr. Krassie Petrova for being my second PhD Supervisor, her valuable input was always welcome.

Thank you to Dr. Alan Litchfield for allowing me to take up space in the AUT Service and Cloud Computing Research Lab that he heads.

I thank Auckland University of Technology for granting me a Vice-Chancellor's PhD Scholarship, without which I would not have been able to embark on this journey. I appreciate the support and management oversight of this PhD that AUT has provided, where the AUT Graduate Research School have been ever present. Thank you, Annalise Davidson and Martin Wilson, from the AUT Graduate Research School for all your help.

There are many other amazing people who have assisted me . . .

Samuel: Thank you for your unconditional support and encouragement throughout my PhD process.

My friend and colleague Gerard Ward: Thank you for your incredible professionality, drive and attention to detail. I benefit from your input always.

I would have abandoned my academic journey over six years ago without the support and logical reasoning of Rachel Cleary, to whom I am grateful.

My friend Jason Wright: Thank you for your help during the entire process. I could not have done this without your continual support.

And, of course, I would like to show my appreciation to my Mother. ☺

I apologize if I have forgotten to include you in this acknowledgement. This thesis would not have become a reality without the active support of many wonderful people I have not thanked here.

# Abstract

The Internet of Things (IoT) has grown from a buzz word into a reality that touches everyone's lives in different ways, from vehicle automation to air conditioning. In this research the question: "What factors improve Risk Maturity Modelling for the Internet of Things?" is used to guide the research. The research problem is the general confusion of terminology and classification of the Internet of Things (IoT) devices and their function in the current literature. Risk identification requires clarity of object definition before the associated risks may be evaluated. Hence, this research builds a semantic engine to broker IoT documents and to specify objects by abstract contextual definitions according to the particular ecosystem. The purpose is to provide business decision-makers with an expert tool for rapid but accurate IoT risk identification. The value of the tool is that the business can apply the tool and determine the risk position without requiring an in-depth knowledge of an IoT device functionality or description, regardless of the device application.

At present the IoT risk context has not been explored in a fashion to establish capability maturity models that suit post event evaluation. In this research the focus is on post event readiness to fill a literature gap that is largely absent from system and device development security literature. The IoT domain is unstable and evolving and the literature is still immature. The observed problem is a lack of appropriate terminology to describe aspects of IoT devices and their functionality, which currently produces a confused mix of semantics. In this research the problems are rationalized into a plan for investigation and the development of a solution. The Design Science methodology is adopted to build a working solution as a prototype for IoT post event risk evaluation. It accepts three inputs that concern the current system state. A semantic engine then processes the three input types and formulates current taxonomies. The capability maturity model then receives the taxonomies and computes the relative maturity levels. This information is a solution to the IoT problem and benefits decision-makers who wish to manage risk and to optimize system forensic readiness.

The deliverable from the research is a prototype instantiation. The prototype takes a selected information input, in the form of a text vocabulary information accumulation. The text input is then parsed through the semantic engine process to provide a risk maturity output. The prototype has been tested manually (Chapter four) on three disparate IoT case studies and then automated (Chapter six) (Maroochy, Target, and Tesla). The

application of the prototype instantiation to each of the three test cases successfully presents a risk maturity analysis. The prototype, as a Proof-of-Concept, demonstrates utility, and is functional as an expert system. It is a sophisticated solution to the problem statement. However, the core of the prototype is a theoretical design principle, which always delivers an unfinished output. Hence, the current research gives starting points for future research and artefact development for commercialization. The Proof-of-Concept output is designed to lay a foundation for future stages of research. The recommendations focus upon new variations in different domain areas, in terms of Proof-of-Value, and the future operational feasibility, in terms of Proof-of-Use. Proof-of-Use will recommend further research into wider generalizations for different IoT domain areas such as the finance sector, the health industry and so on. Recommended future research into Proof-of-Value is toward the functional development of iterative enhancements, investigating specifications for practical use, specifically targeting workplace outcomes and commercialization opportunities.

# Publications

Antony, B. (2020). Containerization: Practical infrastructure and accessibility efficiency for the Virtual Learning Environment. Pacific Journal of Technology Enhanced Learning, 2(1), 41-41.

Antony, B., Cusack, B. (2018, 4-5 December 2018). Is working with what we have enough: The impact of augmented reality on digital evidence collection. Proceedings of the 2018 SRI Security Congress, Perth, Australia

Antony, B., Cusack, B. (2018, 21-23 August 2018). Developing Secure Networks for IoT Communications. Proceedings of the 2018 Cyber Forensic and Security International Conference, Kingdom of Tonga, pp. 217- 228.

Antony, B., Cusack, B. (2018, 21-23 August 2018). Evaluating Network Tools Error Rates for Compliance Reporting. Proceedings of the 2018 Cyber Forensic and Security International Conference, Kingdom of Tonga, pp. 109-116.

Antony, B., Sundararajan, K., & Cusack, B. (2017). Protecting our thoughts. Digital Forensics Magazine (32), 5.

Antony, B., Cusack, B., Ward, G., & Mody, S. (2017, 5-6 December 2017). Assessment of security vulnerabilities in wearable devices. Proceedings of the 2017 SRI Security Congress, Perth, Australia.

Antony, B (2016, 21 July 2017) Presentation: “Forensic Evidence Requirements”, AUT Winter Research Series 2017, 21 July Seminar

Antony, B (2016, 22 July 2016) Presentation: “Layer 2 Forensic Capabilities”, AUT Winter Research Series 2016, 22 July Seminar

Antony, B & Cusack, B. (2016: 30 May). “Technical Report of MS 100”, To Masking Networks INC. USA. 9 pages.

# Awards

2018: Best Forensic Paper, Cyber Forensic and Security International Conference (2018 CFSIC)

2017: Vice-Chancellor's Academic Scholarship

2017: ESET NZ (Chillisoft) Top Scholar Prize

2017: 1st in Graduate Year, Master of Information Security and Digital Forensics

2017: Dean's award for excellence in postgraduate study

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| API | Application Programming Interface |
| CAN | Controller Area Network |
| CMM | Capability Maturity Model |
| DS | Design Science |
| DSRM | Design Science Research Method |
| ERA | Excellence in Research for Australia |
| FTP | File Transfer Protocol |
| HVAC | Heating, Ventilation and Air Conditioning |
| ICS | Industrial Control System |
| IEEE | Institute of Electrical and Electronic Engineers |
| IoT | Internet of Things |
| IS | Information Systems |
| ISN | International Serial Number |
| ISO | International Organization for Standardization |
| MM | Maturity Model |
| NIST | National Institute of Standards |
| NLTK | Natural Language Tool Kit |
| NTSB | National Transportation Safety Board |
| PhD | Doctor of Philosophy |
| PLC | Programmable Logic Controller |
| POS | Point of Sale |
| RAM | Random Access Memory |
| RTU | Remote Transmission Unit |
| SAE | Semantic Analysis Engine |
| SCADA | Supervisory Control and Data Acquisition |
| ZDO | ZigBee Device Object |

# Chapter 1
# INTRODUCTION

## 1.0 INTRODUCTION

The Internet of Things (IoT) has grown from a buzz word into a reality that touches everyone's lives in different ways, from vehicle automation to air conditioning. In this research the research question: "What factors improve Risk Maturity Modelling for the Internet of Things?**"** is used to guide the research. The focus of the research is the addition of capability for evaluation of risk in the IoT context before and after events occur. The purpose of a capability model is to provide metrics that indicate preparedness for service delivery. Higher numbers indicate more robust processes for delivering greater consistency and effectiveness for services, and the lower values indicate a requirement for quality improvement to achieve greater process consistency. At present the IoT risk context has not been fully explored in this fashion to establish capability maturity models that suit post event evaluation (Alaba, Othman, Hashem, & Alotaibi, 2017; Al-Fuqaha, Mohammadi, Aledhari, Guizani, & Ayyash, 2015; Bello, Zeadally, & Badra, 2017; Borgia, Gomes, Lagesse, Lea, & Puccinelli, 2016). In this research the focus is on post event readiness because it is an area that has not been well covered in the literature and is largely absent from system and device development. The research design is summarized in Figure 1.1. It shows the three information inputs: Literature, system and device assessment, and environment/context factors. The core process functionality is a semantic engine, and the output is continuously updated taxonomies so that the capability maturity model can compute for decision making metrics. In this fashion the IoT in its various contexts becomes manageable and assessable for business purposes, and the related risks controllable.

The IoT domain is unstable in many respects for informed decision-making. The literature is still immature, the evaluation of systems and devices is ad-hoc, and the specification of definitions variable (Datta, Da Costa, Harri, & Bonnet, 2016; Gazis, 2017; Mouzhi Ge, Bangui, & Buhnova, 2018; Kovacs & Csizmas, 2018). In Figure 1.1 the problems are rationalized into a plan for investigation and the development of a solution. The three inputs on the left-hand side concern: (1) the current state of IoT standards, security and forensic literature; (2) the ability to evaluate the risk of a system or device; and (3) the immersive IoT context that has many environmental variables. The core development is a semantic engine to process the three input types and to formulate and to update taxonomies. The third element is the capability maturity model that receives the taxonomies and computes the relative maturity levels (Kerrigan, 2013; Balaji, Ranjani, & Geetha, 2019; Carina Sofia & Maribel Yasmina, 2017; Carmela, Carlo, & Domenico, 2006; Dharmpal, 2017). The capability maturity model is taken from the 4-dimensional model known as the "Oxford Model" because it is relevant and responsive to the context (GCSCC 2014). Other traditional capability models lack flexibility for new contexts and comprehensive adaptive features. They tend to force their process organization onto a context rather than being responsive to selecting processes appropriate to the context.

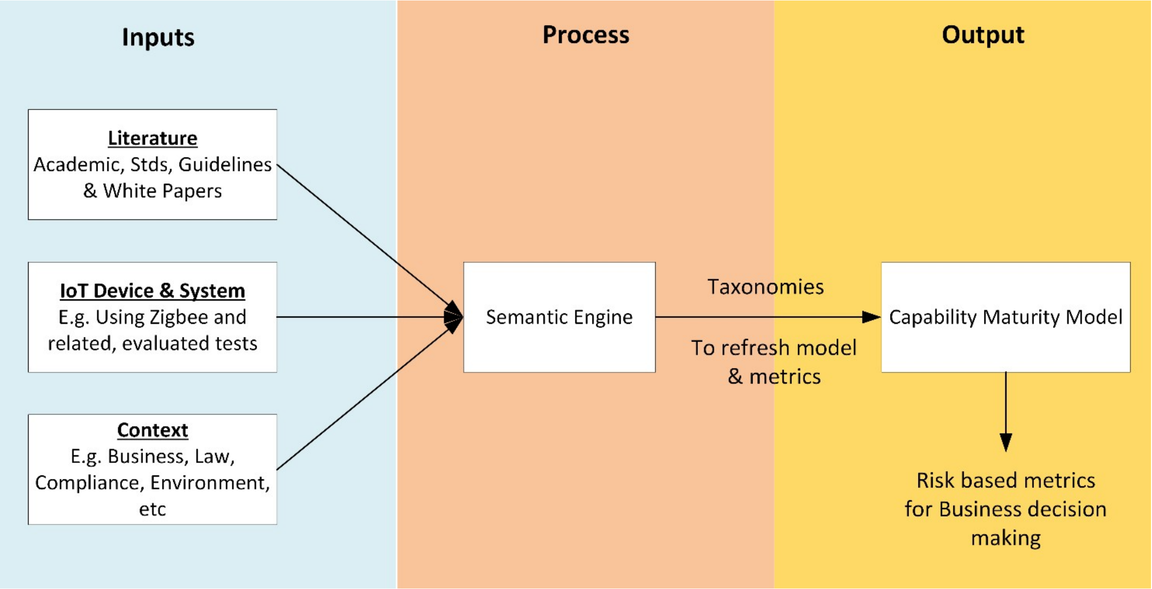

Inputs
Process
Output
Literature
Academic, Stds, Guidelines & White Papers
IoT Device & System
E.g. Using Zigbee and related, evaluated tests
Context
E.g. Business, Law, Compliance, Environment, etc
Semantic Engine
Taxonomies
To refresh model & metrics
Capability Maturity Model
Risk based metrics for Business decision making

*Figure 1.1: The research solution design*

Chapter 1 begins with Section 1.1 and introduces the factors that motivated the research reported in this PhD study. It also presents the potential benefits from the research. Section 1.2 has the research problem, the research question, the research sub-question, and the hypotheses that are addressed. The objective is to fill a gap in knowledge by designing a working solution. Section 1.3 presents the methodology that is developed from the design science guiding literature. Section 1.4 provides a summary of the findings and conclusions that are found in full detail in Chapters 6 – 8. Section 1.5 provides an overview of the structure for the thesis.

## 1.1 MOTIVATION FOR RESEARCH

A general confusion of terminology and classification of Internet of Things (IoT) devices and their function presented the researcher with several problems when attempting to identify risk within the IoT domain. These problems provide the motivation to the researcher to develop and perform the research. The first motivating factor is a scholarly requirement to provide a definition of terms for the IoT domain. This is to assist with IoT risk identification. The second motivating factor is to design a structured and orderly, classification construct, in the form of a taxonomy. The third factor that motivated the researcher is finding solutions for creating an effective taxonomy. The integration of Design Science (DS) research process, which begins through inference, and proceeds through several input-process-output stages, inspired the researcher to produce DS output artefacts by design processes for problem solutions (Gregor & Hevner, 2013; (Venable, Pries-Heje, & Baskerville, 2016). The entire work from the first theoretical stage of an inference artefact, to the final stage to produce an instantiation artefact of the Risk Maturity Model, was inspirational to the researcher, and drove the research forward.

The researcher's primary motivation stems from the difficulty within the IoT domain when tasked to provide a risk evaluation process. The observed problem is a lack of appropriate terminology to describe aspects of IoT devices and their functionality, which currently produces a confused mix of semantics. The researcher was inspired by Herbig (2014, p.32), who provides a statement attributed to Socrates as a truism: "*The beginning of wisdom is the definition of terms.*" The difficulty presented in many research areas is defined as an attempt to dispel jargon, or inferior substitute definitions and terminology (Herbig, 2014). If there are contradictions and inaccuracies apparent when providing the definition of terms within any discipline of thought, it becomes difficult, if not impossible, to provide accurate assessments. These difficulties inspired the researcher to develop, as part of a research process, a definition of terms to form the classification design for IoT infrastructure component designation. The benefits of the potential output are that the maturity model will allow the researcher to enumerate security risk and therefore analyze the maturity level of forensic capabilities within a selected domain or environment. Therefore, the primary motivation for the researcher is to be able to accurately define an IoT object, in order to identify the object risk profile.

The researcher then became passionate about the classification of existing knowledge within the IoT domain (Mayer, Hodges, Yu, Kritzler, & Michahelles, 2017 Jusko, Rehak, Stiborek, Kohout, & Pevny, 2016; Lade, Ghosh, & Srinivasan, 2017;

Mayer, Hodges, Yu, Kritzler, & Michahelles, 2017; Ö, Ajmeri, & Singh, 2016; Patel, Ali, & Sheth, 2017; Prakash, 2016; Sheth, 2016). This began with a systematic description of the investigated objects and devices and culminated with an organizational structure. The process provided a basis of a taxonomy development method, and the subsequent delivery of a taxonomy. The advantage of a functional taxonomy at a domain level is that the taxonomy offers an ordered and logical reference which can be used by researchers to advance knowledge within the taxonomical scope (Ali, Wang, & Haddad, 2015; Andruszkiewicz & Hazan, 2018; Barzegar & Shajari, 2018; Bordea, Buitelaar, Faralli, & Navigili, 2015; Dash et al., 2018; Kairaldeen & Ercan, 2015; Klapaftis & Manandhar, 2013). The researcher was motivated by the benefits of a comprehensive taxonomy for an Information Technology (IT) threat taxonomy. A threat taxonomy presents a hierarchy of threat event groups that have increasing levels of detail, and function to maintain a common lexicon between disparate organizations. IT management professionals rely upon the threat taxonomy to perform risk analysis evaluation. Therefore, the researcher was stimulated by the possibility of designing such an artefact for definitions and categorizations of IoT risk, and process maturity measures. The benefit of the research is for decision-makers requiring evidence for mitigation response. Thus, the establishment of a comprehensive taxonomical structure as an essential precursor to effective risk maturity evaluation adds another motivating factor for the researcher. This is due to the ever-changing nature of the IoT domain, and therefore the incentive for the researcher is to provide an easy-to-follow maturity model which is based upon a relevant semantic determination. The researcher's motivation is to address the problematic nature of IoT risk determination where there is continual change, and there are currently few standardized reference documents.

## 1.2 RESEARCH PROBLEM

The research problem addressed comes from the literature review process which has identified a gap that exists within the domain of Risk Management when applied to the IoT. In order to manage risk, a management process is described as part of the guidelines presented in ISO 31000:2018. The process guideline for managing risk begins with establishing a risk context. In this case it is risk within the IoT domain (Arunadevi & Perumal, 2016; Bekara, 2014; Brumfitt, Askwith, & Zhou, 2015; Ding, Yan, & Deng, 2016; Olivier, Carlos, & Florent, 2015; Suo, Wan, Zou, & Liu, 2012). The establishment of the risk context provides the scope of the research and guides the literature review. The identified gap is within the three risk assessment process steps identified in clause five of the guidelines: Risk identification, Risk analysis and Risk evaluation. The observed problem identified through the literature review process is that there appears to be no single cohesive reference regarding IoT device functionality available that is acceptable to all stakeholders and developers. These three process steps of identification, analysis and evaluation of risk are difficult to perform without an agreed reference frame. Without one, significant variations can occur, and results cannot be easily compared.

Therefore, the research problem is the variability in assessing risk for common objects within the IoT domain. The proposed solution to the research problem is a process that will take an input of domain specific literature, process the information through a semantic analysis engine, and stabilize the problem area with a comprehensive taxonomy mechanism. The automated taxonomy outputs will be processed further to provide input to a maturity model that has a risk enumeration tool. Business decision-makers can then use the standardized outputs and evaluate their current risk maturity level within the IoT domain. The target of the research is to provide a business organization a reference tool that will enable the business to efficiently improve, and therefore gain, their target IoT risk level.

The research output will present a prototype solution to the identified research problem with an expert tool that can be applied by business decision-makers to identify their risk position in terms of maturity. The value of the tool is that a business can apply the tool and determine the risk position without requiring an in-depth knowledge of an IoT device functionality or description, regardless of the device application. As the output is presented in the form of a maturity scale, the business decision-maker will then be able to implement an evidential response with a risk treatment vector suitable to the business's situation. The value to organizations is that the tool provides a self-assessment risk enumeration process that is based upon current guidelines, best practices, standards and information contained within the current literature (see Appendix B). The organization will benefit from the information contained within the tool, and the only requirement is information about how to apply the tool. Therefore, minimal investment is required to use vast amounts of IoT domain specific knowledge.

## 1.3 RESEARCH METHODOLOGY

Design Science (DS) is adopted to guide this research because DS is able to provide problem solution-based designs for artefact/system improvement. Each DS artefact produced as an output can be subjected to an improvement process based upon expert feedback and testing process input. DS produces several artefacts that will each inform progressive stages of the DS activity application (Finelli, Borrego, & Rasoulifar, 2015; Gregor, 2006; Gregor & Hevner, 2013; Gregor & Jones, 2007; Herbig, 2014; Menard, Bott, & Crossler, 2017). The DS activities in this research conclude by producing a scaled maturity model as the final instantiation output. The use of case study research (three diverse instances) is integrated into the research methodology to provide research rigor and validity. The application of the DS research processes in this research is intended to provide artefact development. The application of the DS processes focus upon the development and design of the problem solutions as artefact outputs (Gregor & Jones, 2007; Hevner & Chatterjee, 2010; Koppenhagen, Gaß, & Müller, 2012; Nunamaker, Briggs, Derrick, & Schwabe, 2015; Nunamaker Jr, Chen, & Purdin, 1990; Offermann, Levina, Schönherr, & Bub, 2009; Peffers, Rothenberger, Tuunanen, & Vaezi, 2012; Peffers et al., 2006; Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007; Vaishnavi & Kuechler, 2015; Venable & Baskerville, 2012). Thus, throughout the DS artefact creation process, knowledge is gained, and the dissemination and disclosure of the resultant knowledge is inherent within the DS research process. The DS process provides a methodology that will explore the selected problem, and a context in which to test the research hypotheses. The DS guidance then provides a process to design and test the artefact output with the inclusion of expert information for design improvement iterations. Finally, DS is used to test the hypotheses, and provide research validation in the form of a final artefact output ready for dissemination and generalization towards other applications.

The first activity within the DS process utilised in this research is initialised with an artefact output that is designed to define the specific research problem investigated. Identification and conceptualisation of the research problem is important because the identified problem complexity provides justification of the value for an effective solution in the form of an artefact. The definition of the value of the solution holds the reasoning underpinning the researcher's designation of the problem's level of importance. The value of the solution also determines the researcher's motivation to deliver the solution.

The objectives for an effective solution are determined, in the form of inferences, from the problem definition. The second DS activity requires an input of knowledge of what is possible, and that which is feasible. The requirement for an effective application of the second activity is an input of knowledge about the state of problems and the efficacy of current solutions, if any. The objective of the second activity step of the DS process, is to theorize objectives and choose desirable ones that are better than current solutions. It includes defining theoretical artefacts that will support novel solutions for the problem.

For the purposes of this research, three case studies are evaluated, to provide a feasibility check, and validation of the steps from inference to theory. The output from each of the three case study evaluations, is presented in Chapter four. The theory output of the second activity presents the input to the third DS process sequence, where the theory input is processed into an application output. The third activity of the DS process is to design and develop the proposed artefact theorised in the second activity. The contribution of the research is embedded in the design of the artefact output of this activity. The fourth activity of the DS process begins with demonstration of the use and application of the designed artefact output from the third DS process activity. The fourth activity will proceed towards a refined, and graduated output. The fifth activity will involve an internal development process that matures the artefact design so the output can be used to inform the creation of the maturity model, as part of the fifth DS activity. The sixth activity of the DS process takes the output from the previous activity, to provide the input of the demonstration of effectiveness evaluation process by the analysis of algorithmic efficiency and application.

## 1.4 RESEARCH FINDINGS

The inference formed during the early stages of the research, that an assessment of manual risk maturity evaluation methods may present a practical solution pathway, enabled the researcher to define the objectives of the solution artefacts and identify the deliverables for the research findings. The deliverables are presented as artefacts, and as output of the overarching Design Science methodology. There are two further findings presented as novel solutions to the problem areas. These two findings are the initial inference, and the use of theory from the Information Systems (IS) data science domain, as an exaptation. The resultant deliverable is a Proof-of-Concept in the form of a software based, algorithmic prototype. The key findings, in the form of DS artefacts present the knowledge contributions from this research. The findings, deliverables, and research contributions are shown in Table 1.1.

The overarching finding of this research, in the form of a comprehensive artefact deliverable, is the Prototype Instantiation. The prototype takes a selected information input in the form of a text vocabulary information accumulation. The text input is then parsed through the semantic engine process to provide a risk maturity output. The manual and automated application of the prototype instantiation to each of the three test cases successfully presents a risk maturity analysis. The prototype demonstrates utility and can be seen to be robust and reliable.

*Table 1.1: Findings, deliverables and research contribution*

| Finding / Deliverable | Research Contribution |
|---|---|
| **Exaptation**<br>Known Solution Extended to New Problems | The exaptation output presents prescriptive knowledge contributions of the software, algorithmic techniques, using Data Science workflow and Natural Language Processing (NLP) design knowledge to the IS problem context. |
| **Inference**:<br>There is a link between cyber forensic analysis and IoT risk aspects | The inference that there is a correlation between cyber forensics and risk identification aspects presents the novel descriptive knowledge conceptual contribution designed to provide a solution to the fully defined problem |
| **Method**:<br>Test case manual risk identification.<br>Validation identification and process testing control | A valuable contribution to research is presented by the method artefact as the method establishes a process sequence that can be refined or adapted for use in other contexts. |
| **Instantiation**:<br>Semantic Analysis Engine | The exaptation of Natural Language Processing (NLP) for use in the application of Semantic Analysis for risk evaluation, presents a novel research contribution. |
| **Construct**:<br>The taxonomy output derived from the semantic analysis process, in the form of a construct artefact designed to inform the Risk Maturity Model artefact | The Taxonomy construct is an artefact formed as an integral component of the Semantic Analysis process. The Taxonomy creation, starting with an input of information accumulation is then subjected to a domain relevant term extraction processes, to output risk attributes. |
| **Risk Maturity Model**:<br>The Oxford maturity model architecture processes are the taxonomy construct to output the final artefact | The risk maturity creation model artefact is a development of the SAE construct artefact, where the model focuses on utilizing the taxonomy process output to inform the risk maturity creation model. |
| **Prototype Instantiation**:<br>The final artefact output is the IoT Risk Maturity Model prototype | The prototype instantiation outcome is developed from an analysis of the application of the method, model and construct artefacts. The prototype instantiation demonstrates a feasible and functional solution to the research problem. |

The Prototype Instantiation deliverable presents recommendations for future research. The nature of the prototype is a theoretical design principle, which is an unfinished output that provides avenues for future research and artefact development. The Proof-of-Concept output of this research is designed to lay a foundation for the future stages of research (Kenneth David, 2015; Louise & Thomas, 2019; Marcello, 2009; Maria, 2019; Olaronke, 2018; Richard, Marco, & Sietse, 2019; Sandro, Lucile, Ludovic, & Bruno, 2017; Troels & Henrik, 2008; Ying, Han Tong, & Wen Feng, 2008). Therefore, the nature of the findings presented in this thesis gives a prototype, theoretical design principle, and an unfinished output that provides motivation for future research, artefact development of iterative enhancements, specifications, and direction for wider generalization. Further research is for design improvements involving more complex tasks, analyzing finer grained metrics, and goal understandings. The Prototype is used to analyze three widely divergent test case studies, presenting generalization for the three IoT domain aspects of each test case.

The Prototype of the Risk Maturity Instantiation is evaluated as a Proof-of-Concept, where the overarching requirements are utility and efficiency. Therefore, the future research recommendations are to evaluate and research the prototype for Proof-of-Value and Proof-of-Use. Research for Proof-of-Value is to develop improvements of the functional quality of both the processes and the technical components, by which the research artefacts presented can create value. The research output will identify new phenomena of interest, their correlates, and present knowledge of the new theoretical logic to explain the observed phenomena. Research into Proof-of-Value will also present rigorous metrics and empirical evidence for solution efficacy. Research for Proof-of-Use is to investigate the knowledge needed for end users to build instances suitable for the user's problem domain, and to generalize solutions. The research advantages of the recommended future research of Proof-of-Use are to intentionally develop classes of knowledge that create business value for problem owners. There is potential for the Proof-of-Use research output to participate in commercialization ventures, which will provide commercialization revenues and deliver resources to advance further research.

## 1.5 THESIS STRUCTURE

The thesis is communicated in eight Chapters. Chapter one presents an introduction and overview of the thesis. It begins with an introduction of the research question that focuses and guides the research. Chapter one then presents the motivation that drives the researcher, an overview of the research methodology and approach, the research objectives and goals, the expected research findings, structure overview, and linkage to the remainder of the thesis.

Chapter two gives a review of literature, which begins with a description of the literature selection process, and then literature concentrating on IoT risk contexts, such as vulnerabilities, exploitation and mitigation vectors. Two examples of IoT physical and network layer device communication pathways are then presented. The knowledge gap is identified, through a process of inference, which is presented as the first artefact output, and a research contribution for the thesis. The absence of post event risk treatments is also noted. The inference artefact, or gap then leads to identifying the research problem which is the definition of terms. The next sections concentrate upon taxonomic development through semantic analysis, and current capability maturity modelling processes. The literature review process also let the researcher read research methodology and the conclusions are presented in Chapter three.

Chapter three presents the research methodology and research process design. It identifies the problem, and discusses taxonomy creation methods, both theoretical and practical as the proposed primary solution to the identified problem. The research design is presented and the specific DS research phases. Including: the hypotheses to be tested, the research sub-questions, and research question to be answered. Finally, the research data requirements, the rationale for inclusion of case studies as a Proof-of-Concept, and the data analysis processes, are given.

Chapter four presents a pilot study and Proof-of-Concept in the form of three case studies. It demonstrates the manual process of taxonomic creation that is used in the research, as applied to each case study. The studies selected involve diverse applications of IoT and cover a wide timeframe from 2000 to 2018. Each case study is presented in turn, starting with the historic ‘Maroochy sewage spill’, then the ‘Target data breach’, and finally a ‘Tesla autonomous vehicle crash’. Each case study presentation follows a similar format which begins with an analysis of the case, and then demonstrates the taxonomic creation process and risk attribution. Chapter four then concludes with an overarching review of the chapter and the links to Chapter five.

Chapter five presents a method variation to the Semantic Engine implementation. The T-SQL component of the semantic engine is evaluated as part of the DS methodology and assessed for utility and functionality. The T-SQL component is seen to be using computing resources beyond expectation and requiring over 99% of the available resources. An alternative, Open-Source solution is adopted, integrating Linux, Python and the Natural Language Tool Kit (NLTK). The variation to the methodology is presented in Chapter five, where the software adaption is explained, and the suitability is outlined. The overarching DS research methodology allows changes to improve the potential output artefacts, without changing the research method.

Chapter six presents the research findings of the Proof-of-Concept prototype. The three sections of Chapter six analyze information from three different IoT case study applications in diverse domains. The three case studies are analyzed by the Semantic Analysis Engine (SAE) individually in each of the sections. Test case one, Maroochy Shire Sewage Spill, test case two, Target Private Information Data Breach and test case three, Tesla Vehicle Automation, are subjected to an automated analysis by the SAE. The three test cases are subjected to the same overarching process steps, in the same order, and generating different output of data. The three sections of Chapter six identify each process step, defining each query, text execution and filter management, by providing an input-process-output logic progression.

Chapter seven discusses the findings presented in Chapter six. The findings from the experiment provide evidence of identifiable risk aspects which, in turn, inform the capability maturity tool. The findings are used to test the two hypotheses presented in Chapter three: **H1**: *Risk aspects are identified using cyber forensic and data analysis techniques;* and **H2**: *The output from Hypothesis One testing informs a Risk Maturity Tool to identify IoT risk.* Chapter seven also answers the research sub-Questions: **RSQ1**: *What risk aspects are identified using cyber forensics and semantic data analysis techniques* and **RSQ2**: *Which risk inputs inform a Risk Maturity Tool for the Internet of Things?* The Main research Question is then answered: **RQ1**: *What factors improve Capability Maturity Risk Modelling for the Internet of Things?* Chapter seven then presents and discusses the key findings, in the form of DS output artefacts. Validity, in the form of evaluation of utility and robustness is discussed, and reliability measures. Finally, the implications and limitations are considered, and a conclusion is made.

Chapter eight concludes the research and begins by presenting the contributions to knowledge from the research and concludes with recommendations for future research. The research output delivers seven novel contributions to knowledge as integral components of the Design Science research process. It has delivered a Proof-of-Concept prototype that is used to process three test case scenarios to output a cybersecurity maturity risk analysis. The Proof-of-Concept output presents a solution to the problem statement. The recommendations outlined are focused upon new variations in different domain areas, in terms of Proof-of-Value and future operational feasibility, in terms of Proof-of-Use. The knowledge gives dissemination information in the form of a variety of exploratory, theoretical, experimental, and applied science deliverables. There is potential of the Proof-of-Use research output to participate in commercialization ventures for commercialization revenues that will deliver resources to advance further research.

# Chapter 2
# LITERATURE REVIEW

## 2.0 INTRODUCTION

The literature review forms an important element of the overarching Design Science methodology adopted for this research. Chapter two describes the process of problem identification, problem enumeration and conceptual framework construction. The initial Design Science artefact output from this research is an inference, that there is a link between cyber forensic analysis, and the identification of the Internet of Things (IoT) risk aspects. The inference presents a novel research contribution, underpinning the basis for the literature review. Thus, Chapter two defines the research problem by identifying an information gap, and then identifies key requirements for potential solutions. The key requirements therefore inform the design and practical aspects of the problem and solution space. The literature selection seeks the relevant justificatory knowledge, best practice guidelines, practical design principles, conceptual constructs that have relevance to the research topic. The design principles facilitate construction and the build of solutions for the research problem.

Chapter two begins with an outline of the literature selection methods in Section 2.1. Then Section 2.2 reviews risk aspects in the IoT environment, and Section 2.3 gives IoT examples. Section 2.4 explores post-event evaluation methods for forensic activity. In Section 2.5 a full evaluation of semantic methods is made that are useful for assessing data inputs and taxonomy outputs. In Section 2.6 capability maturity creation methods are reviewed and in Section 2,7 cross construct communication issues are identified.

## 2.1 LITERATURE SELECTION

The selection of literature has been identified as an important task when assessing the fundamental nature of the theory within the discipline of Information Systems (Gregor, 2006). Basic questions that are applied to identify the building-blocks are placed into a number of classes. The classes selected as relevant for the proposed research relate to these topics: Domain, Structure, Epistemology, Stakeholder and Operational. As the proposed research will be focused on the intersection of human behavior and physical object properties, it is beneficial to draw knowledge from the artificial world of constructs, the social world of best practices and industry guidelines, and the legal world of regulations and standards.

Thus, the basic theoretical underpinnings of Taxonomical construction begin with the selection of appropriate domain knowledge. This allows the scope of the literature to remain practicable, and most importantly, relevant. In this research, the cyber-physical literature is selected from a body of knowledge contained within Excellence in Research for Australia (ERA) peer reviewed, A* ranked journal publications. The research incorporates information published within the IEEE Internet of Things Journal, ISSDN 2327-4662, as the journal is not only A* ranked (as ranked by the ERA journal ranking system) but is also co-sponsored by the IEEE Sensors council, the IEEE Communication society, the IEEE Computer society and the IEEE Signal processing society. This journal will provide appropriate material that will become the primary literature input that will be processed by the Semantic Engine.

The best practice and industry guideline literature inputs will be selected from respected sources such as National Institute of Standards and Technology (NIST), and the International Organization for Standardization (ISO) (Ross, Feldman, & Witte, 2016; Stoneburner, Goguen, & Feringa, 2002; Stouffer, Lightman, Pillitteri, Abrams, & Hahn, 2014; Voas, 2016; Jeffrey Voas, Feldman, & Witte; Weiss et al., 2015). The Legal literature will be sourced from relevant legal and legislative sources such as the New Zealand Legislation Database. There will also be industry and business specific information sought from experts in business on best practices. Figure 2.1 shows an overview of the literature input integration depicting the data collection and aggregation from the multiple but domain specific sources, which is then subjected to appropriate term selection / extraction processes. These terms will then be used to generate a viable taxonomy. Indicators are utilized to assess maturity and provide actionable insights which will be used to provide a basis for the Maturity Model.

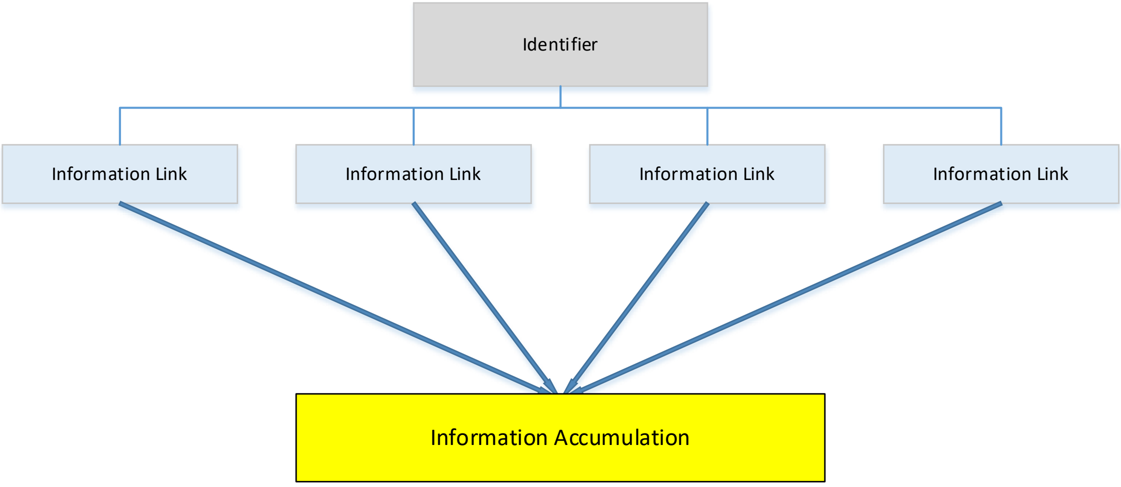


***Figure 2.1: Overview of literature input integration***

## 2.2 RISK CONTEXTS

Risk represents potential points of failure in a system. In the following sub-sections potential points of failure in IoT network security, exploitation vectors, mitigation, and architecture, are reviewed.

### 2.2.1 The Context of Network Security for IoT

Protecting all parts of a complicated modern network has become difficult when developing a security strategy. Modern networks have not only become complex but through the integration of extranet equipment and external componentry such as e-commerce servers and remote access services, the security has become porous (Botta, de Donato, Persico, & Pescapé, 2016; Mengmeng Ge, Hong, Guttmann, & Kim, 2017; Li, Han, & Jin, 2016). This is where the network perimeter, which consists of firewalls and application gateways, can no longer be tightly controlled. A breach of an organization's security, where the integrity of the data is compromised by attacks, can often have ramifications that go beyond the organization's border (Lee, Geng, & Raghunathan, 2016). The ability to utilize advanced information technologies and communications has altered an organization's landscape, where collaborative projects of significantly expanded scope and scale are initiated and managed from various points both within and without an organization's border (Moody, Kirsch, Slaughter, Dunn, & Weng, 2016).

***Table 2.1: Network vulnerability exploit risk vector***

| Attack | Definition |
|---|---|
| Denial of Service (DOS) | Network rendered inaccessible through large volumes of traffic generated to crash servers or overwhelm routers and firewalls. |
| Scanning | Network is examined for vulnerabilities by scanning for open ports and information obtained through listening on the ports that have been left open |
| Spoofing | Packet header manipulation to change IP information to indicate the packet has been sent from an IP address other than the true origin |
| Routing | Permits network data to be routed through a specific point on the network, overriding routing decisions |
| Protocol | Vulnerabilities in existing protocols such as HTTP, DNS and CGI exploited. Can include system and software exploits such as buffer overflows and unexpected input errors. |

### 2.2.2 The Context of Vulnerability and Exploitation Vectors

How an intruder gains entry to a computer network and what actions the intruder invokes can be termed an attack type. Some common forms of attack types and corresponding exploit risk vector are shown in Table 2.1 (Bhuyan, Bhattacharyya, & Kalita, 2017); Hansman & Hunt, 2005; Wei, 2012).

### 2.2.3 The Context of Vulnerability Mitigation

A developing model of attack mitigation, especially when there are requirements for scalability and ensuring quality of service levels, is to manage information and network services at the border of a network. There are other ancillary advantages to implementing border management at the network edge such as: increasing service response time through latency reduction, limiting the scope of potential data spread and increasing efficient network resource consumption. Precise application of the edge management concept can reduce occurrence of single failure points, thereby increasing robustness across the system as a whole (Sicari, Rizzardi, Miorandi, & Coen-Porisini, 2017).

However, this ideal is difficult to realize when integrating an IoT component into an analysis of vulnerability or risk mitigation. The difficulty lies in the porous nature of border delimitations within the IoT environment of interconnectedness. The initial research inference artefact is that there is a link between cyber forensic analysis and the identification of the Internet of Things (IoT) risk aspects. Therefore, a solution that will enhance risk vulnerability identification and enumeration is to analyze cyber forensic investigations of exploitable risk vectors (Saarikko, Westergren, & Blomquist, 2017; Shahzad, Kim, & Elgamoudi, 2017; Trappey, Trappey, Hareesh Govindarajan, Chuang, & Sun, 2017).

### 2.2.4 The Context of IoT Architecture Security Variations

The emergence of new revenue stream generation from smartphone applications and services can be viewed as the original driving force behind the IoT technology explosion. To take advantage of the potential profits, new technology in the form of smart sensors were developed to support the smart services offered by smart phones. This, in turn, developed new paradigms which stimulated rapid IoT development (Bello, Zeadally, & Badra, 2017). The new paradigms range from communication protocols such as Bluetooth Low Energy (BLE), Wireless Fidelity Direct (WiFi Direct) and Near Field Communication (NFC) to Intelligence based networks such as Software Defined Networks (SDN) Information Centric Networking (ICN) and Network Functionality

Virtualisation (NFV). These paradigm changes have resulted in the flood of smart applications and their ancillary connected sensor devices.

Network infrastructure, extending to and including the Internet has been expanded to connect sensor devices, such as heartbeat monitors, with operations and health care organizations. The devices are often self-configuring and intelligent, interrelating in a dynamic global infrastructure. The Internet of Things (IoT) is formed when these devices communicate across networks via the Internet Protocol (IP) and form a ubiquitous and pervasive worldwide network. The devices are individually identifiable through unique addressing, which is a requisite condition to achieve network communication (Yan, Zhang, & Vasilakos, 2014; Zarpelão, Miani, Kawakani, & de Alvarenga, 2017).

The ideal of these connections is to enhance the quality of life through the interaction of the devices with the operations (Mashal et al., 2015). The stated ideal can be as simple as a power meter interacting with electrical power provider who does not have to rely on a meter reader to provide the consumer accurate and up to date billing information. The example demonstrates an enhanced quality of life through a cheaper power supply due to less overhead and increased accuracy as well as the timely nature of the information gathered. There has been an expansion of devices that integrate sensors that connect to the Internet to allow remote human interaction with advanced smart service provision as the goal (Botta, de Donato, Persico, & Pescapé, 2016). There are concerns identified because the devices or sensors can collect information without human direction and the data gathered may be analyzed and mined for information about human social life and activities, which conveys perceptions of uncertainty and risk (Yan, Zhang, & Vasilakos, 2014).

IoT network systems consist of diverse entities acquiring data from their operating environment and interacting amongst themselves (Janczewski & Ward, 2019). Large volumes of data are often sent via wireless communication links, and as such, the data is vulnerable, and at risk of being captured, modified or otherwise manipulated. Methods and procedures for access control and authentication are essential to preserve data integrity and therefore to promote IoT system expansion and security endorsement (Sicari et al., 2017). However, the lack of cross-platform modelling and communication link protocol differences, impede the implementation of effective access control and authentication methods. As demonstrated in Section 2.6, the communication protocol differences between the two communication systems outlined show that a single security approach is problematic.

Another factor identified that inhibits the maintenance of security and privacy in the IoT paradigm is the enormous volume of data that is generated by IoT devices (Mineraud, Mazhelis, Su, & Tarkoma, 2016). The problems range from determining the owner of the data, which then translates to access and control right assignations. This is at the heart or the CIA security triad when determining security for the transfer of access control when the data moves through the various nodes and other control mechanisms. IoT devices, whether actuating or sensing, utilize different patterns when transferring data and employ communication protocols differently. This causes difficulties when implementing security policies. The problem expands when attempting to provide a forensic analysis of events after the fact, as the raw data will not be stored, due to the large data storage requirements.

Within large-scale network control systems, the inclusion of hardware devices that are widely available and are readily obtainable, has increased design efficiency and developmental flexibility, especially when investigating device interoperability (Genge, Graur, & Haller, 2015). However, the expansion of prevalent hardware, along with associated software brings higher levels of risk due to the same factors of accessibility that provides the benefits of efficiency and interoperability. The risk that the ubiquitous devices have been investigated by malicious actors is high and must be taken into account, particularly when the infrastructure within which the device is incorporated is critical. Nevertheless, there are potential benefits with ubiquitous devices that drive their use.

## 2.3 PHYSICAL AND NETWORK LAYER COMMUNICATION EXAMPLES

The IoT communications infrastructure relies heavily on layer 1 and 2 (See Appendix A) to achieve efficient data transfer (Bello et al., 2017). The goal is to achieve seamless communication amongst uniquely identifiable devices with network connectivity that may be connected directly to the Internet. An example of network connectivity is the Controller Area Network (CANbus) standard that has been adopted by the automobile industry, where sensor communication is local to the vehicle itself without utilizing a local host computer. This is in comparison to ZigBee protocol for low cost, very low powered devices, which provides two-way device to device communication. This gives the ability for the device to join a network, interact and pair with other devices without outside control or configuration, across disparate networks via Internet connectivity.

### 2.3.1 CANbus

The CANbus system is investigated as an example of a well-defined highly evolved IoT protocol with many advanced features such as error checking, polling, arbitration and transmission capability removal capabilities (Iqbal et al., 2017). A Controller Area Network (CAN) bus is a two-wire system, originally developed by Bosch to send many short messages utilizing a message broadcast system, designed to replace complex wiring harnesses prevalent in existing motor vehicles. CAN then became an International Standardization Organization (ISO) defined standard, ISO-11898:2003, which defines the originally designed two wire serial communications bus. There are integrated features specified, such as self-diagnosis and data error repair as well as high immunity to electrical transmission interference (Suresh, Daniel, Parthasarathy, & Aswathy, 2014; Tewari & Gupta, 2018; Weber & Studer, 2016; Wolf & Serpanos, 2018).

The CAN communications protocol portion of ISO standard ISO-11898:2003, defines the information / data transference rules between devices on the CAN bus network (Anusha, Senthilkumar, & Naik, 2017). As can be seen in the Figure 2.2 below, the lower two layers of the Open Systems Interconnect (OSI) model are employed, with transmission, signaling and some levels of control utilizing the physical or lowest layer of the OSI stack. Thus, the communication protocol must use polling and activity detection in order to send and receive data, and as such is a carrier sense, multiple access, and collision detection protocol. There is also message priority polling and arbitration.

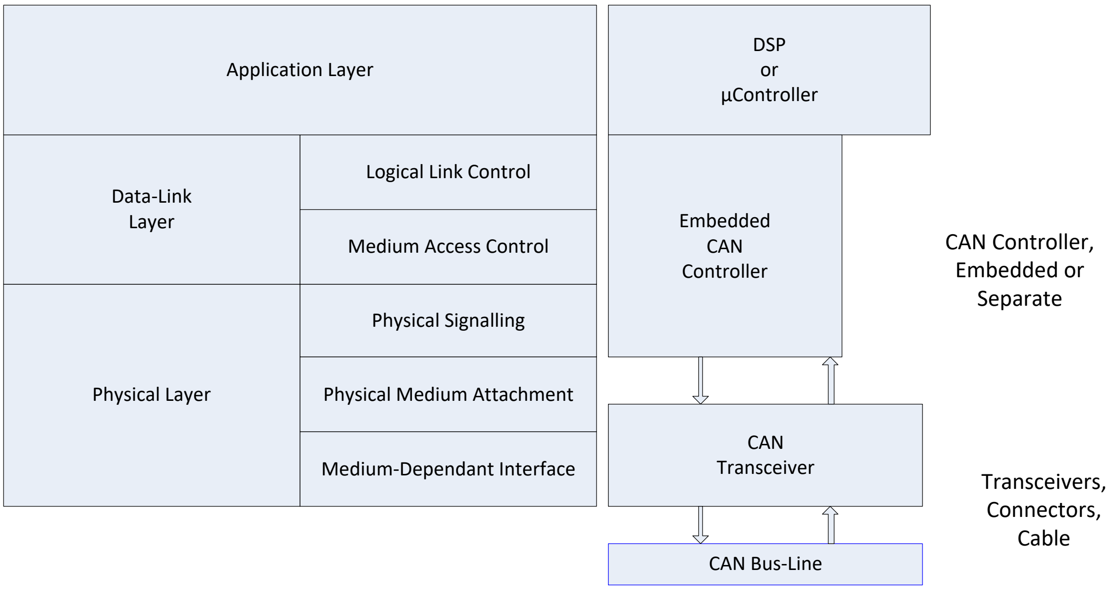


***Figure 2.2: CANbus stack***

The abbreviation for this is CSMA/CD+AMP. CSMA (Carrier Sense Multi-Access) indicates that a period of inactivity must occur before and sensor on the bus will attempt to send a message. There is a priority identifier pre-programmed, so bit-wise arbitration can resolve collision issues, where the highest priority always wins access to the bus, called Collision Detection and Arbitrated Message Priority (CD+AMP).

Interestingly, as the communication on the bus is multicast in the last logic identifier which has the highest priority and as such will keep on transmitting. The standard CAN protocol employs an 11-bit unique identifier, allowing 2048 unique message identifications which has been extended to provide a 29-bit identifier, which can provide 537 million unique message identifications. This translates to 2048 unique devices on a standard CANbus and 537 million devices on the CANbus with the extended protocol parameters. There is provision in the standard for deferential balanced signaling which presents high fault tolerance and robust noise immunity compared with twisted pair cabling structures. The standard provides strong error checking procedures, incorporating five different error checking methods: two at the bit level and three at the message level. An error frame is generated by the receiving node if a message fails any of these five error checking methods and is placed on the bus. This forces the original transmitting node to continue resending the message until it is received correctly. Should an error message limit be reached, the transmitting node will be removed by its controller. This allows the ISO 11898 standard to utilize a high speed with a maximum signaling rate of 1Mbps along the bus length, but with a limitation of 30 nodes.

This shows that Forensic Capabilities can be implemented, and data captured is assignable to uniquely identified devices contained within the CANbus system being investigated. Thus, the Cyber Forensics Capability can be seen to be mature. This is confirmed through the high level of data captured in the third test case analyzed manually in Section 4.5. The information presented to the National Traffic Safety Board (NTSB) was forensically evaluated after the incident occurred without having to add any specific Cyber Forensic data capture commands. Whilst it would be challenging for a business enterprise to have the expertise on hand to analyze each risk aspect of the CANbus system, a categorization of Cyber Forensic analyses can provide risk enumeration.

### 2.3.2 ZigBee

ZigBee provides the tools for IoT system and device evaluation. Analysis of the ZigBee protocol starts with the ZigBee transmission format designed for wireless media. The ZigBee specification is based upon the IEEE 802.15.4 technical standard which defines the operation of Low-Rate Wireless Personal Area Networks (LR-WPANs). The IEEE 802.15.4 standard specifies the Media Access Control (MAC) and Physical Layer (Layer one) parameters, and ZigBee extends the standard by providing upper layer descriptions, not specified in 802.15.4 (Al-Fuqaha, Mohammadi, Aledhari, Guizani, & Ayyash, 2015). Thus, the 802.15.4 determines the usage of the wireless physicality such as channel selection, signal management and energy functions. There have been continuous developments of the 802.15.4 standard with additions over time such as channel hopping to support industrial applications, different frequency ranges for different countries and utilization of Direct Sequence Ultra-Wideband (UWB) and Chirp Spread Spectrum (CSS). The 802.15.4 standard also determines the MAC layer transmission of MAC frames through the physical wireless channel (Lopez, Rios, Bao, & Wang, 2017). This level of control offers a management interface, self-managing physical channel accessed and network beaconing. It is important to note that these frames are not exchanged as standard Ethernet frames. Figure 2.3 shows the interaction between the lower two layers of the OSI stack and the layers above layer 3 when considering the interaction outlined in 802.15.4.

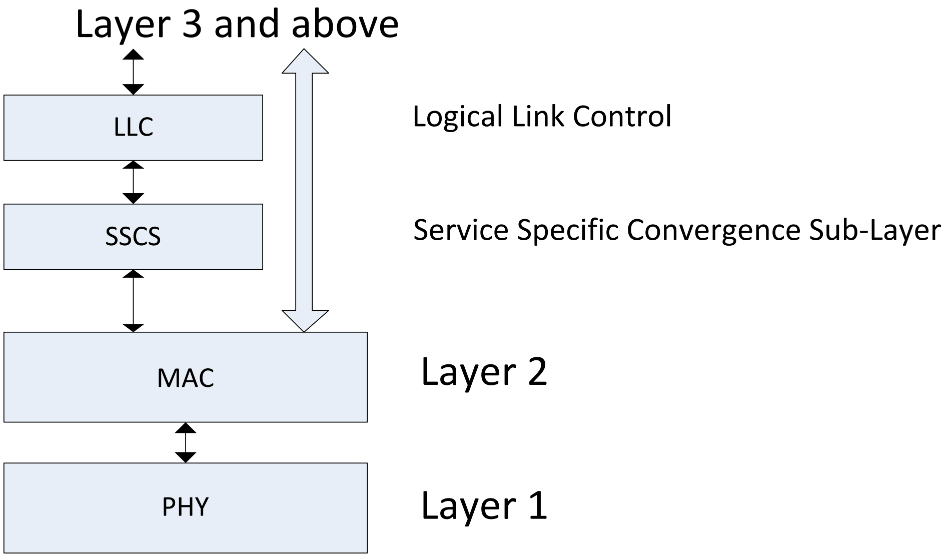


***Figure 2.3: 802.15.4 Standard OSI interaction***

An issue with IoT smart device connectivity is the requirement for all devices to be uniquely identified. ZigBee messages are transmitted through a combination of two types of dynamically assigned 16-bit addressing. The first, which uniquely identifies up to 65,536 ZigBee physical nodes as a member of a Personal Area Network (PAN), is then combined with another dynamically assigned 16-bit device address, giving a total of over four billion uniquely identifiable ZigBee transmission addresses. These addresses then utilize a radio transmission channel, and each node must be registered on the same channel in order for transmission of data to occur. ZigBee radios automatically negotiate channel selection from 12 channels. Each ZigBee radio is also assigned a permanent and unique 64-bit serial number. Therefore, a boundary is created through the use of channel selection, with sub-boundaries created through the application of the dynamically assigned PAN identification addresses. Finally, the individual nodes are identified through the dynamic assignment of the 16-bit node identification address. Figure 2.4 shows the utilization of the addressing parameters and demonstrates how the addressing can be used to provide communication security. A primary perimeter of defense is formed through channel selection, and secondary perimeters are established through PAN ID allocation. The final defense perimeter is defined through the use of Address identification.

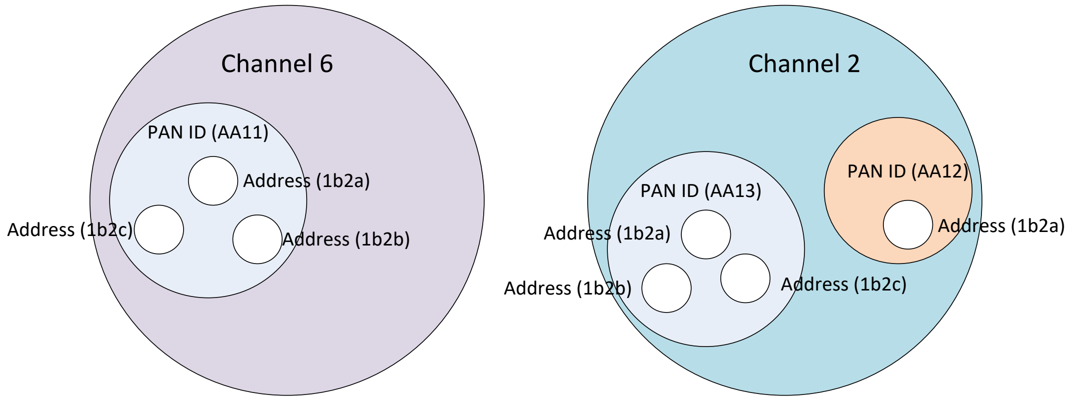


***Figure 2.4: ZigBee boundary delimitation***

In much the same way as Ethernet is a protocol that is used to reliably transmit useful data, or payload, through Ethernet Frames, ZigBee utilizes its own protocols, such as the ZigBee Application Programming Interface (API) to transmit and receive programming communication. The goal of the API protocol is to enable the reliable and predictable transmission of highly structured data. The ZigBee Frame structure contains sub-structures designed to encompass the different types of sensor and actuator data through the use of more than 20 different types of API frames. The different frame types are identified through the use of a Frame Type ID contained within the first four bytes of the transmitted data. The first four bytes also contains information about the frame's start byte, the length of the frame as well as the frame type identification. Thus, the ZigBee communication structure provides an OSI layer three network level specific to ZigBee. This layer then communicates to the upper layers of the ZigBee protocol stack, as defined within the ZigBee Alliance Specifications as shown in Figure 2.5.

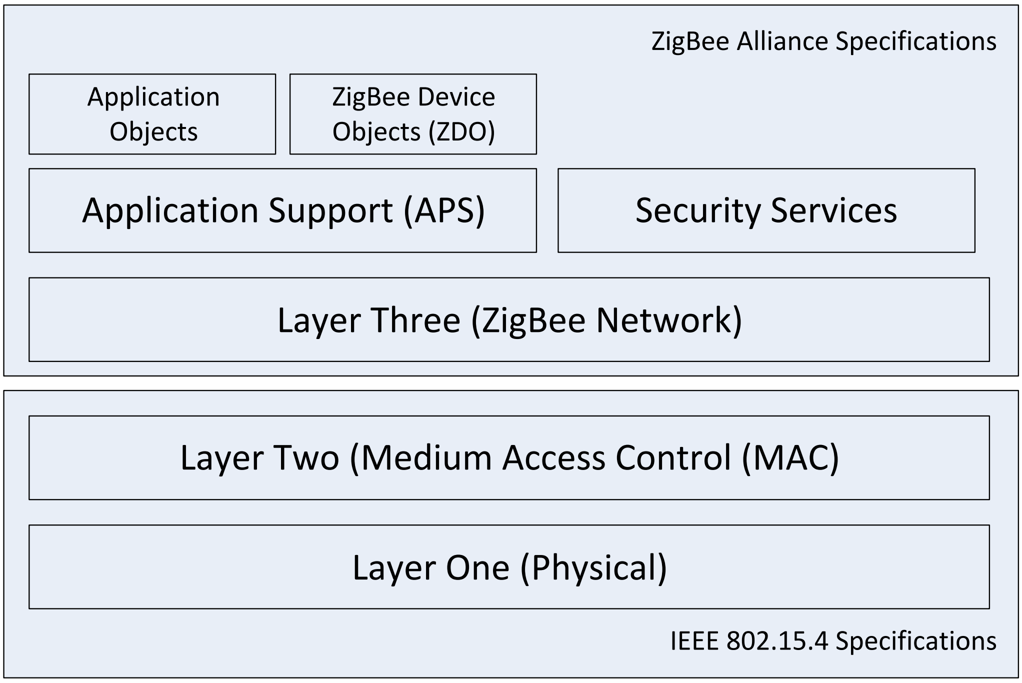

ZigBee Alliance Specifications
Application Objects
ZigBee Device Objects (ZDO)
Application Support (APS)
Security Services
Layer Three (ZigBee Network)
Layer Two (Medium Access Control (MAC)
Layer One (Physical)
IEEE 802.15.4 Specifications

***Figure 2.5: ZigBee protocol stack.***

The Application Support Sub-layer (APS) describes specific task messaging that allow ZigBee devices created by different manufacturers to communicate and co-operate seamlessly. The ZigBee Device Object (ZDO) layer is used by the radio itself. The ZDO layer provides capabilities for network management, device and service discovery that include network routing requests and end-point information. Therefore, ZigBee application profiles have been developed which are collections of protocols and common definitions whereby ZigBee devices are designed to work together in particular environments. These include physical applications such as: Health care, Home automation, Telecommunication services and Smart energy installations. The self-configuration and standardized messaging capabilities, allows seamless communication over network topologies such as ZigBee mesh, providing robust node communication by relaying transmissions through intermediary nodes. ZigBee mesh topology provides reliability through the ability of the network to function even with multiple node failure by using alternate paths to maintain a connection.

The auto-configuration and standardization capabilities present security issues and challenges, however. Security considerations are an essential component when communicating across a network, especially when utilizing wireless transmission. An information gap has been identified as the IoT community is yet to develop standardized security protocols across all IoT devices (Mineraud, Mazhelis, Su, & Tarkoma, 2016). However, the boundary delimitation described above provides border definitions and demarcation and demonstrates strength through defense in depth, whereby each node is protected by a PAN addressing system, and the PAN is in turn protected through radio transmission channels, as shown in Figure 2.6. However, it is still theoretically possible for a malicious actor to establish the physical transmission details and collect transmission data payloads and actors can even perform ‘Man in the middle’ attacks. The issue arises when standard network security implementations such as data encryption are utilized when investigation IoT sensors and devices in terms of resource overhead. It is important to also consider the additional development costs which will need to be incorporated in the end user cost of each device.

The defense in depth security does mean, however, that forensic capabilities are also difficult to implement. With the assignation of identical node identification addresses that are separated by the PAN ID and the Radio channel, it is difficult to record complete and uniquely identified data that can be forensically examined and used as evidence. The addition of a Cyber Forensics System, as shown in red in Figure 7, that records information from Layer Three, The Security Services, Application Support and the ZigBee Device object, would increase the Cyber Forensics Maturity of the ZigBee Alliance Specifications.

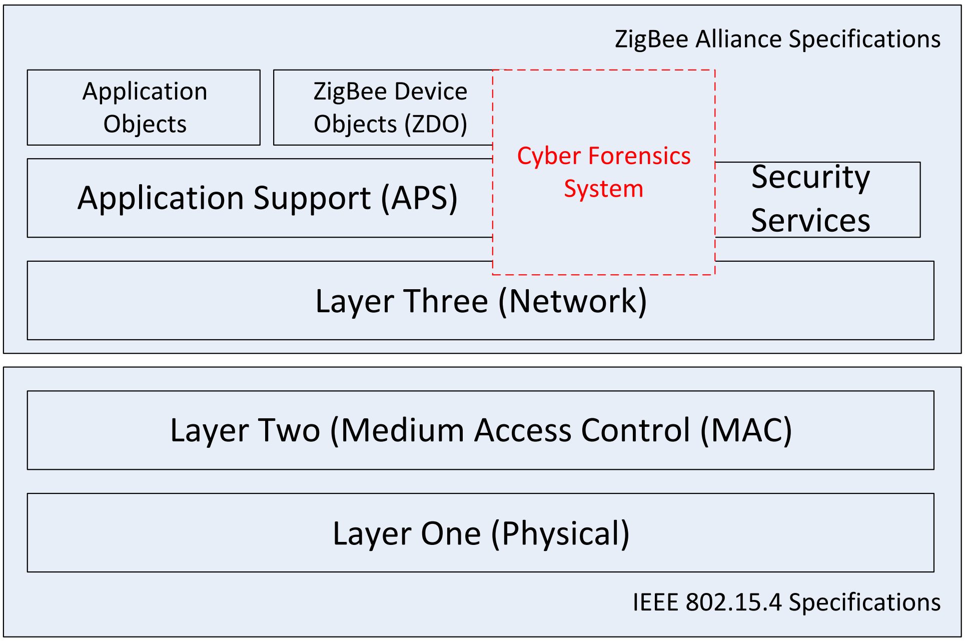

ZigBee Alliance Specifications
Application Objects
ZigBee Device Objects (ZDO)
Cyber Forensics System
Application Support (APS)
Security Services
Layer Three (Network)
Layer Two (Medium Access Control (MAC)
Layer One (Physical)
IEEE 802.15.4 Specifications

*Figure 2.6: ZigBee protocol stack with conceptualised forensics system*

The integration of Cyber Forensic capabilities presents challenges to a business enterprise. The core business of the enterprise does not include a knowledge base of the technical depth required to apply analysis techniques to enumerate risk within IoT ZigBee communication. However, the analysis of Cyber Forensic information is accepted at court, along with best practices and international standards for compliance. These also exhibit novel risk categorisation opportunities. Therefore, Section 2.5 and 2.6 demonstrate technical challenges that business enterprises encounter when enumerating risk, and the potential that Cyber Forensic analysis presents.

## 2.4 POST EVENT EVALUATION

Figure 2.7 identifies the positioning of a Cyber Forensic system that facilitates post event evaluation of an incident. In the following sub-sections, the issues of pre- and post-event evaluation are reviewed so that the distinction between the two is clear. The topics of traditional network security, protecting networks, security system for the IoT, and digital forensic capability are reviewed (Borgia, Gomes, Lagesse, Lea, & Puccinelli, 2016; Botta, de Donato, Persico, & Pescapé, 2016; Mengmeng Ge, Hong, Guttmann, & Kim, 2017).

### 2.4.1 Fundamentals of Traditional Network Security

Traditional network security is based upon the ability to configure perimeter defense control where enterprise assets are utilized inside a clearly defined delineation. Any access originating beyond the perimeter could be identified and therefore blocked, which is effective when dealing with external attacks. The identification and blocking process is controlled by firewalls and Intrusion Detection Systems (IDS) which are predominately rules based. A set of preconfigured parameters or rules are parsed in order to manage each incoming packet that is compared to a single rule-set and the packet is then manipulated according to the comparison result. The manipulation is either to pass on to the next rule comparison, deny, or drop. There are only so many actions that can be evoked with a rules-based system, which is also static, due to being pre-configured, and the rules are not able to be automatically updated by environmental network changes (Li, Han, & Jin, 2016; Lopez, Rios, Bao, & Wang, 2017; Mashal et al., 2015; Mineraud, Mazhelis, Su, & Tarkoma, 2016).

An analogy of this form of traditional network security is the medieval castle and has been referred to as “defense in depth.” The defense in depth analogy is seen with the castle walls separating insiders (with access permissions and authorization) and outsiders (all the rest). The access was controlled through a single city gate, with the castle guards inspecting each individual along with the incoming goods and packages. The network equivalent to this process is the traditional firewall. The defense process is only possible with a clearly defined security perimeter with a single ingress point with which to control access.

Thus, network security is a process of detecting, and then preventing unauthorized external intrusions. The detection and prevention process can be likened to a battle over control and ownership of data (Figure 2.7).

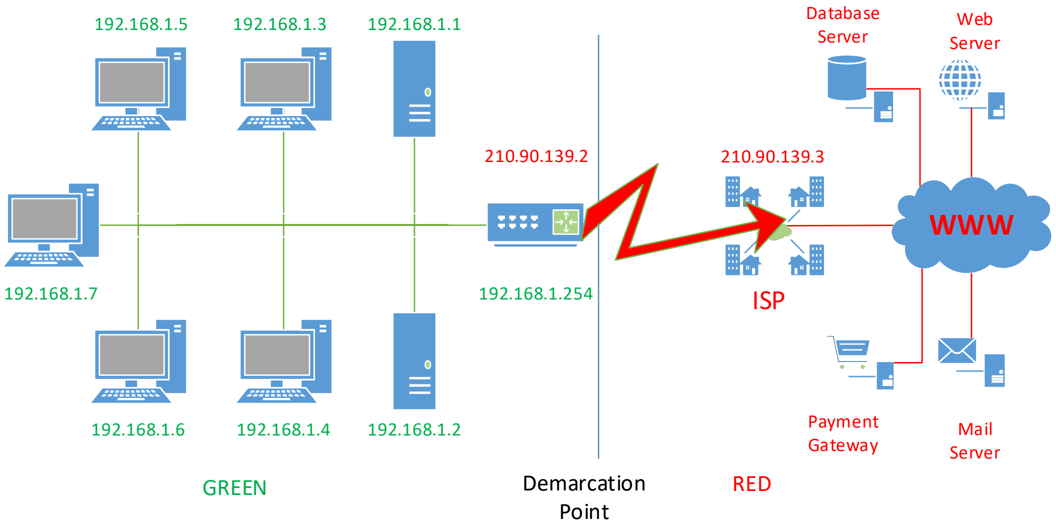


*Figure 2.7: Red-Green zone concept*

### 2.4.2 Creating Secure Networks

Security priorities are diametrically opposed, whereby one essential component is accessibility or data availability and the other two are data confidentiality and data integrity. Thus, the more available the data, the less secure, and the more secure the data, the less the data is available. It is important to note that confidentiality is not security but is a sub-set of security. Security goes beyond system function, data protection and inappropriate use prevention.

Another important facet of security is non-repudiation. This is where an action can be assigned to a unique actor, and is often overlooked when considering security, especially when focusing on the CIA triad. The objective when providing non-repudiation requires proof of both the integrity and the origin of data. This proof must be asserted with high assurance.

### 2.4.3 Input Network Security as a System for IoT

The context and environment of IoT requires an in-depth risk analysis. In order to commence building an IoT network security model, a conceptualization process is required. The process of conceptualization applies to several principles from the complex system of network security. The goal of the conceptualization process is to provide a common frame of reference that defines the system being examined, which will then develop into a discussion of the management of the system being abstracted (Suresh, Daniel, Parthasarathy, & Aswathy, 2014; Tewari & Gupta, 2018; Weber & Studer, 2016; Wolf & Serpanos, 2018). The Oxford dictionary online defines the word system as "a set of things working together as parts of a mechanism or an interconnecting network; a complex whole." (https://en.oxforddictionaries.com/definition/system). Thus, in the examination of data being transmitted upon a complex interconnected and interrelated network, the use of systems tenets assists to identify and control the system. As a system usually comprises a set of related components formed to achieve a certain task, the analysis begins with a walk through of the process whilst attempting to fulfill objectives and identifying constrains and enumerating resources. The scrutiny of the process will form a regular subordination of objects resulting in a presentation of essential principles and facts. Thus, the process described is systems analysis, with the goal being to determine how the system works to lay the groundwork for suggested improvements. Functions must be broken down into their smallest parts but is a holistic process to look at the system as part of an environment of external and internal influences.

It is important to note within information systems security theory, that only the information that is unknown is relevant, and everything else is extraneous, and therefore can be ignored (Shannon, 1948). The ability to remove immaterial information exemplifies a fundamental concept when comprehending systems, that an all-inclusive viewpoint is not required, and only full understanding of the non-redundant parts is required. Only the information that can affect the system as a whole is relevant, everything else can be removed. Therefore, information systems analysis is not a process of reductionism, where the system of components and functions must be broken down into their smallest parts, but is rather a holistic process, looking at the system as part of an environment of external and internal influences (Grzenda, Awad, Furtak, & Legierski, 2017).

As the conceptualized network security system is part of a wider convergence of interaction, both positive and negative, the process of analysis of network security as a system must therefore begin with the identification of a border or boundary. Thus, the network security system is a function of its border, whereby constrains can be best described only after the definition of the network border of influence. The network influences information security within the network border delineation, and is in turn, influenced by events form beyond the network border. The concept of a boundary when analyzing a network system can be arbitrary and difficult to accurately determine, especially when integrating IoT devices. The border is not easily established through identifying a demarcation point between the system and its environment, such as the physical point where Internet connectivity is supplied through an appliance, such as a firewall. Visualizing a Local Area Network (LAN) as being separate and distinct from the Internet is no longer viable, and they are not two discrete systems, as the boundary has become permeable with information data flowing back and forth through the border control appliance to and from Internet connected devices. As the system being analyzed interacts dynamically with the surrounding environment, which itself consists of many systems, and hence, the network concept becomes a matter of perspective and changing scale.

In order to define the system as a whole, it is not necessary to describe the minutiae for each component part or process step. A system can be understood with a definition of its environmental interaction, the border or boundary and the systems input and output parameters. As part of a system, input is information received from the environment, throughput is the changes caused by the system and output is the information that leaves the system and is sent back into the environment. Thus, a holistic viewpoint can be upheld, and a black box approach can be taken, where and output can be viewed in terms of an input, rather than having to define or even examine the throughput. The modular approach of using black box concepts reduces the reliance on technical ability which in turn reduces complexity and can simplify the analytical process.

The challenge with the black box modular approach is that the analysis of a network security system is both complex and adaptive, and taking an approach that is overly simplistic may cause spurious results. The challenge can be mitigated through the application of a hierarchical approach to systems analysis (Bélanger, Cefaratti, Carte, & Markham, 2014). At a higher, or macro level, the system can be viewed as an abstraction of the system as a whole which may be over generalized. The lower or micro level involves a detailed analysis of the component parts and functions, but the understanding of how the discrete parts fit together can become lost. Using a single approach will produce limitations, thus it is a combination of both these concepts that produce the best results when analyzing a system. Another concept in using a hierarchical approach brings a sub-system affect to the overarching system, which in turn has an effect upon the internal sub-systems. Therefore, a hierarchical perspective as a comprehensive tactic for model construction encourages analysis of data from multiple levels to avoid incompleteness (Zhang & Gable, 2017).

A security analogy of the hierarchical, modular system as defined by its border is to consider a cell within the human body as part of a biological system (Pronk, Pimentel, Roos, & Breit, 2007). The cell is defined by its border or cell wall. The cellular wall / border cannot be absolute, or the cell would perish through lack of input of nutrient proteins and the ability to output waste. Thus, whist the cell is delineated from its environment by its border, the border must be semi-permeable or porous in order to provide functionality by staying alive and interacting with the surrounding environment. The cells surrounding environment, the body, can be considered the parent system, and the cell can be considered the sub-system. Both the sub-system and the parent system input nourishment and output waste through transformation activities. These activities can be viewed as black box or modular processes and do not need to be examined in detail. It is sufficient to realize that the systems, both sub and parent promotes growth and sustainability, whilst also providing energy to repair damage. The use on analogy in this instance is valid as one of the primary precepts of systems theory is the assentation that there are principles of organization that are universal for all systems. Therefore, these principles are used to understand and manipulate these systems and to provide better understanding of the system being analyzed.

### 2.4.4 Digital Forensics

Post event analysis is undertaken using the tools and techniques of digital forensics. Section 2.6 considers the integration of widely varied and seemingly disparate devices into the IoT ecosystem and the potential for communication of forensic information between areas. Also concern that forensic capability is available for post event investigation is raised. Whilst a standard factor when considering security vulnerability mitigation is to partition the communication network and establish borders. The creation of borders is to establish controls for communication between the zones, as discussed in Section 2.2 where the concept of red versus green zone communication control is introduced. The potential for forensic information gathered within the zones, to be utilized beyond the firewall border to establish proactive forensic readiness into a zone outside the established perimeter.

Digital Forensics is the processes for gathering, preserving, analysis, and reporting digital evidence that may be used for system improvement or legal proceedings. Lutui (2016) further categorized the scope of Digital Forensics into different areas (Figure 2.8). Security sets up protective layers to defend assets from harm, but the cost of perfect security is either too high or impossible. Hence, Digital Forensics is a further risk mitigation strategy for when security provisions prove inadequate to protect assets. The readiness for investigation includes evidence retention capability and evidence acquisition skills. These provisions are not always factored into security designs that focus resources on protection in the belief it will be adequate.

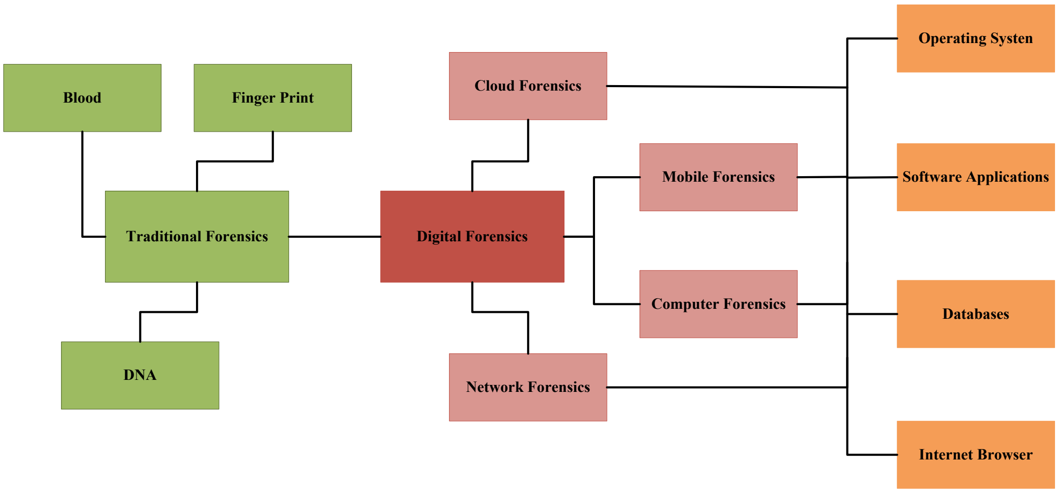


*Figure 2.8: Digital Forensic Areas*

Post-event preparation and understandings are underdeveloped for the IoT. The IoT requires conceptualizing in some of the ways outlined in the previous sections. The IoT may not have firewalls as networks do, so new architectures require exploration for effective post-event risk mitigation. In this research the proposal is made for post-event mitigation by preparation, and by design. The prototype artefact is to elaborate the inference between the IoT and risk. By its very nature the artefact is an expert tool (Appendix B) for simplifying IoT understandings, and for delivering useful information ahead of any incident. A business manager with little or no IoT risk knowledge can be fully informed as to the risk an IoT device or service presents to a business environment. The forensic capability is also assessed.

## 2.5 DATA EVALUATION

Central to the ability to manage IoT risk assessment and to evaluate all IoT data is the formulation of relevant and applicable taxonomies. In addition, the capability for semantic analysis operationalizes the taxology processes and connects the logical and physical worlds of the IoT context. In the following two sub-sections these matters are discussed.

### 2.5.1 Taxonomy

A taxonomy provides a structured and orderly, systematic classification construct, and is designed to organize a subject in a particular way, leading towards an understandable tree structure of a subject. For the purposes of this research, each level of the taxonomy will focus on a smaller and smaller physical subset in the IoT arena (Andreas, Erik, Wei Lee, Isam, & Stuart, 2012; Anthony, Karthik, Kulathur, & Gregg, 2013; Thuraisingham, Tsybulnik, & Alam, 2008, Greenidge & Hadrian, 2011). The intended outcome is to produce a logical structure that will assist in the classification of the wide variety of documents to be used in this literature review. Therefore, a hierarchical arrangement of terms that have been developed and presented in order to organize the concepts itemized within the documentation search. Thus, the taxonomy presented is the means by which the vast amount of information reviewed can be navigated and the results refined.

To better understand and describe the shared set of concepts which represents the current knowledge specifics within the IoT domain, a formal and explicit specification, or ontological tool is needed. The ontological structure proposed is designed to facilitate understanding of a comprehensive set of IoT models and the relationships between these models. Therefore, a formal naming approach is undertaken, where definitions of entities, the interrelations, properties and types will be described and itemized. Ontologies, however, imply that the scope of information may be broad, and envelop several and different taxonomies (Benites & Sapozhnikova, 2013, Greenidge & Hadrian, 2011; Chun-Che & Hao-Syuan, 2011).

An ontology can therefore be described as a fully faceted taxonomy with no unresolved ambiguities. The production of an effective and appropriate taxonomy leads to an effective ontology, once the concepts have been fully developed and described. A properly styled taxonomy will produce an ontology that is both powerful and unambiguous (Harleen, Ritu, & Alam, 2011, Marcialis, Roli, Coli, & Delogu, 2010, Pulkkis, Grahn, & Karlsson, 2008). As the objective of this research is to resolve ambiguity and to clarify questions, which is the key to good information access, the development of an effective taxonomy will be the main research output.

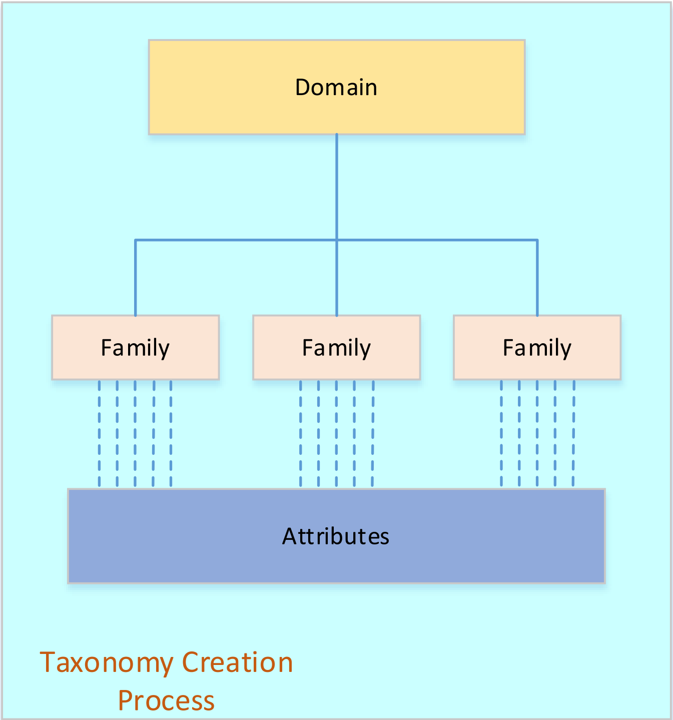


***Figure 2.9: Taxonomy creation process***

The prime directive of effective taxonomy creation in this research, is to design a hierarchical tree structure that will capture a set of interrelationships in the IoT area. It is important to ensure that the taxonomy created relates to the same category of knowledge and is therefore orthogonal. To facilitate the ontological design requirement, a faceted navigation structure is designed, where the intention is to produce a hierarchical taxonomy that has been categorized and therefore normalized. Figure 2.9 shows the faceted structure that builds an improved model of the IoT domain literature contributions and therefore produces a more evolved conceptual framework. Thus, the taxonomy shown in figure 2.8 contain a top-level domain node which represents different attribute or contexts within each identified family node.

#### 2.5.1.1 Taxonomy Development

In order to gain a more complete understanding of the IoT security variations, a formal definition of the IoT domain from several professionally respected sources is required. This research investigates security information flow, both inside and beyond communication boundaries, and therefore explanations and classifications for the IoT's architectural requirements will form the basis of a common vocabulary that is used throughout this research. The definitions have been taken from professional institutions such as: National Institute of Standards and Technology (NIST), Institute of Electrical and Electronics Engineers (IEEE), and International Organization for Standardization

(ISO). The process of definition is an important step of Taxonomical creation and will be used extensively when configuring the Semantic Analysis Engine. An example of the process performed manually is demonstrated next.

#### 2.5.1.2 Manual Analysis Example

There are several NIST publications that cover the IoT domain. These publications include domains such as NISTIR 8062, and internal report entitled: “An introduction to privacy engineering and risk management in federal systems”. Also, NISTIR 8063, an internal draft report entitled: “Primitives and elements of internet of things (IoT) trustworthiness, and NIST SP800-183 a special publication entitled: “Networks of Things” (NIST & Voas, 2016).

NIST SP800-183 “Network of Things” begins by stating that there is a lack of formalized description of the components that direct IoT trustworthiness, operation or lifecycle. The SP800-183 document investigates what science underpins the IoT, if any, and puts forward the underlying and foundational IoT science. The model proposed does not include definitions of exactly what is or is not a ‘thing’ but rather considers the behavior of a ‘thing’ with regard to effects on the flow of work and data within the networked environment.

NIST SP800-183 clarifies an important foundational point, that there are two separate paradigms being investigated, the ‘Network of Things’ (NoT) and the ‘Internet of Things’ (IoT) and characterizes the IoT as a subset of the NoT. The IoT is the NoT with internet connectivity and communication paths. Industrial control systems are part of a delineated physical level arrangement, such as a Local Area Network (LAN). The importance of this delineation is expressed when comparing various types of NoT, as there are many varieties differentiated by IoT. It is important when considering cases from different domains such as Vehicular, Medical and Critical infrastructure applications. It is common to define NoT and IoT interchangeably, however the difference in terminology provides a subtle but essential differentiation when investigating security and forensics at and beyond the firewall.

The SP800-183 document begins by describing the most basic of building blocks, with which more complex and complete systems can be constructed, evaluated and compared, as ‘primitives.’ Providing a description at such a low level focuses on establishing a vocabulary of definitions which classifies the characteristics of these basic parameters. The vocabulary unification and integration are designed to facilitate the exchange of information amongst networks designed for different purposes. Thus, the

SP800-183 document delivers a model that clarifies the foundational aspects of IoT, whereby the components that express the behavioral aspects of IoT are provided, rather than supplying IoT definitions (NIST & Voas, 2016).

The components of primitives as expressed within NIST SP800-183, form the taxonomical derivation, defined through parameters of Description, Properties, Assumptions, General statements, and Risk identification. The parameters are used to create the hierarchical structure of the taxonomy entities for the manual analysis examples shown. Table 2.2 shows each parameter and lists the parameter definition used in each of the manual examples. The ensuing Tables from 2.3 to 2.8 are reproduced for their vocabulary and definitional attributes for later reference in this research.

***Table 2.2: Parameter definition for manual analysis***

| Parameter | Parameter definition |
|---|---|
| Description | Defines the behavioral aspect of the NoT unit. |
| Properties | Describes the NoT unit functional properties. |
| Assumptions | Explains information on operational use and application. |
| General Statements | Lists overarching information of use. |
| Risk Identification | Identifies Risk attributes. |

***Table 2.3: Identification of IoT sensor primitives***

| Primitive | Sensor |
|---|---|
| Description | A sensor is an electronic utility that measures physical properties. |
| Properties | • Sensors are physical.<br>• Sensors output data. |
| Assumptions | • Sensors may transmit device identification information.<br>• Sensors should have the capability to supply authentication information.<br>• Sensors may have multiple recipients for its data.<br>• The frequency with which sensors release data impacts the data relevance.<br>• Sensor precision may determine how much information is provided.<br>• Sensors may transmit data about the health of the system. |
| General Statements | • Humans are not sensors.<br>• Humans can, however, influence sensor performance. |
| Risk Identification | • Security is a concern for sensors.<br>• Reliability is a concern for sensors |

*Table 2.4: Identification of IoT aggregator primitives*

| **Primitive** | **Aggregator** |
|---|---|
| Description | An aggregator is a software implementation based on mathematical function(s) that transforms groups of raw data into intermediate, aggregated data |
| Properties | • Aggregators may be virtual.<br>• Aggregators require processing 'horsepower.'<br>• Aggregators have two actors for consolidating large volumes: Clusters and Weights. |
| Assumptions | • Sensors that communicate with pother sensors may act similarly to aggregators.<br>• Aggregated data may suffer from information loss due to rounding and averaging. |
| General Statements | • For each cluster there should be an aggregator.<br>• Aggregators are either event driven or act at a specific time for a specific period.<br>• Some NoT instances may not have an aggregator. |
| Risk Identification | • Security is a concern for Aggregators.<br>• Reliability is a concern for Aggregators. |

*Table 2.5: Identification of IoT cluster primitives*

| **Primitive** | **Cluster** |
|---|---|
| Description | A Cluster is an abstract grouping of sensors that can appear and disappear instantaneously. |
| Properties | • Clusters are abstractions of a set of sensors along with the data they output.<br>• Clusters may be created in an ad hoc manner or organized according to fixed rules.<br>• Clusters are not inherently physical. |
| Assumptions | • Clusters may share one or more sensors with other clusters.<br>• Clusters are malleable and can change their collection of sensors at any time. |
| General Statements | • The composition of clusters is dependent on what mechanism is employed to aggregate the data. |
| Risk Identification | • The mechanism impacts the purpose and direction of a specific NoT |

*Table 2.6: Identification of IoT weight primitives*

| Primitive | Weight |
|---|---|
| Description | Weight is the degree to which a particular sensor's data will impact an Aggregator's computation. |
| Properties | • Weight may be hardwired or modified on-the-fly.<br>• Weight may be based on a Sensor's trustworthiness.<br>• Different NoTs may leverage the same sensor data. |
| Assumptions | • Different NoTs may re-calibrate the weights to comply as per the purpose of a specific NoT |
| General Statements | • It is not implied that an Aggregator is a functionally linear combination of sensor outputs.<br>• Weights can be based logical insights.<br>• Weights will affect the degree of information loss during the creation of intermediate data.<br>• Repeated sampling of the same sensor may affect that sensor's weighting. |
| Risk Identification | • Security for weights is related to possible tampering of the weights.<br>• The correctness of the weights is crucial for the purposes of a NoT |

*Table 2.7: Identification of IoT communication channel primitives*

| Primitive | Communication Channel |
|---|---|
| Description | A communication channel is a medium by which data is transmitted. |
| Properties | • Communication channels move data between sensing, computing, and actuation.<br>• Data moves to and from intermediate events at different snapshots of time.<br>• Communication channels will have a physical or virtual dimension. |
| Assumptions | • Communication channel dataflow may be unidirectional or bi-directional.<br>• No standardized communication channel protocol is assumed. |
| General Statements | • Communication channels may be wireless.<br>• Communication channel trustworthiness may cause sensors to appear to be failing, when it is actually the communication channel that is failing. |

| | |
|---|---|
| | • Communication channels can experience disturbances, delays, and interruptions. |
| Risk Identification | • Redundancy can improve communication channel reliability.<br>• Performance and availability of communication channels will greatly impact any NoT that has time dependent decisions.<br>• Security and reliability are concerns for communication channels. |

*Table 2.8: Identification of IoT decision trigger primitives*

| Primitive | Decision Trigger |
|---|---|
| Description | A decision trigger creates the final result(s) needed to satisfy the purpose, specifications and requirements of a specific NoT |
| Properties | • A decision trigger is a conditional expression that triggers an action.<br>• A NoT may, or may not, control and actuator via a decision trigger.<br>• A decision may have a binary output, but the output may be a continuum of output values.<br>• A decision trigger may have a built-in adaptation capability as the environment element changes. |
| Assumptions | • A decision trigger will likely have a corresponding virtual implementation.<br>• A decision trigger may have a unique owner. |
| General Statements | • Decision trigger results may be predictions.<br>• A decision trigger may feed its output back into the NoT creating a feedback loop. |
| Risk Identification | • Failure to execute decision triggers in in a timely manner may occur due to tardy data collection, inhibited sensors, low performance Aggregators and sub-system failures.<br>• Decision triggers act similarly to Aggregators and may be considers as a special case of Aggregator.<br>• Security is a concern for decision triggers.<br>• Reliability is a concern for decision triggers. |

### 2.5.2 Semantic Analysis

Manually generated taxonomies incur high workload overhead and remain incomplete and present customization difficulties (Zafar, Cochez, & Qamar, 2016). Inadequacies have also been acknowledged through the identification of relevant information when relying upon keyword searches. This is demonstrated when attempting to filter large datasets to produce taxonomies that are relevant to the domain being investigated

(Thangaraj & Sujatha, 2014). However, semantic based taxonomies provide structured knowledge which can in turn be applied to information retrieval (Liu, Song, Liu, & Wang, 2012).

Automated and computer assisted systems are part of every field and growing rapidly (Sadeghian et al., 2018). An example of this growth trend is shown within the legal domain (Branting 2017). The design of intelligent systems has shown extensive research designed to address challenges in expense and time required to extract information from text. Special emphasis is placed on information extraction as the information is of special importance from a legal perspective. However, the task can be processed efficiently, since almost all the information in this domain comprises natural human language. Information retrieval techniques aid in the existent automation processes utilized for creating and displaying meaningful citation networks.

However, as only a small subset of the vocabulary or text accumulation is relevant to the citation network use, the determination of relevance is a key aspect of the analysis task. Visualizations are challenged when presenting a comprehensive view of entire citation networks. The subsets of a citation network which are relevant depends both on rules, as the attributes of nodes and citations, as the attributes of edges. A vocabulary subset relevant to a public health domain would include both nodes defining the powers and duties (e.g., doctors, epidemiologists, coroners) and citations indicating the relative authority. The section of a statutory framework that is relevant consists of the vocabulary subset given by nodes and edges demonstrating a semantic relationship to the domain.

Maxwell, Antón, Swire, Riaz, & McCraw, (2012) developed a system to help software companies attain regulation compliance. They study the taxonomy of legal-cross references in the acts related to financial information systems and healthcare, identifying examples of cross references contained within legal texts indicating conflicting compliance requirements. Maxwell et al. (2012) obtains several cross-reference types by determining cross reference patterning occurring within case studies analysis: *constraint, exception, definition, unrelated, incorrect, general*, and *prioritization.* They postulate that this set of labels are generalizable to other legal domains. An example is shown with the similarity between the terms limitation and constraint, which can be generalized to laws governing software systems. However, Maxwell et al (2012) indicate that potential quality improvements can be gained through including additional data from best practices and international standards as specialist sources.

De mAAT, Winkels, & van Engers, (2009) created an analysis engine that achieved high accuracy when extracting and resolving references within Dutch law, after studying the structure of references A classification system was applied that divided the classifications into five categories: Normative, Metanormative, Delegating, lifecycle and Informative. An automatic, highly accurate analysis engine was designed and demonstrated to extract text and resolve legal citations (Sadeghian et al., 2018). The reference of legislation to case decision forms the basis of the evaluation. Case studies were manually evaluated, and the extracted patterns presented the data that informed clustering processes. The artefact output presented learning methods that promised future research direction. The learning methods did not rely upon human annotative input nor predefined sets of labels. Similar works have analyzed interlinked actions of emergency response within public health systems (Sadeghian et al. 2018). The authors analyze the nature of organizational links to provide characterization. For example, a section of text can be defined as an 'action' which defines these links as 'legal mandate', and then refine the process utilizing information gathered from previous process applications.

#### 2.5.2.1 Process Steps

There are two main steps in the process of creating a domain specific taxonomy. The terms that are appropriate to the domain under investigation need to be identified, and these terms must then be placed in a hierarchy. This process then forms the basic taxonomical structure when using semi-supervised construction methods (Kozareva & Hovy, 2010). The domain specific terms can be manually identified but are also extracted from bodies of published work. The most common form of automatic taxonomy creation is through rules-based methods, where certain patterns are established, which can then be used to deduce a hypernym-hyponym relationship. The relationship is shown when 'a [[NOUN A]] is a [[NOUN B]]' pattern is found, NOUN B can be surmised to be a hypernym of NOUN A (Sang, Hofmann, & de Rijke, 2011). An elementary example of this relationship, in the IoT domain is demonstrated when the following 'noun pairs' are is found in literature: 'a Temperature Transducer [NOUN A] is a Sensor [NOUN B],' and a Smoke Detector [NOUN A] is a Sensor [NOUN B].' Thus, the extrapolation of the hypernym-hyponym relationship is that Sensor is a hypernym of both Temperature Transducer and Smoke Detector. Thus, taxonomically, the following can be inferred: Sensors are a high-level category or classification which contains the following 'sections' Temperature Transducer and Smoke Detector.

A further, deeper semantic analysis of the above also produces the following relationship (Figure 2.10): Transducers and Detectors are also a ‘sub-section’ of Sensors and may be structured as an intermediary classification between Sensor and Smoke detector and Temperature Transducer. Thus, the following, and more complete hierarchical structure is developed:

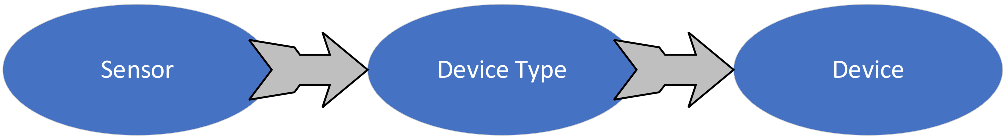


***Figure 2.10: Classification process***

***Table 2.9: Domain specific relationship***

| **Domain** | **Sensor** |
|---|---|
| Device Type | Transducer<br>Detector |
| Device | Temperature<br>Smoke |

Further semantic analysis conducted on bodies of published literature ascertains that ‘a Transducer is something that converts energy into another form, and that ‘a Detector converts real world conditions to an analog or digital representation’ as shown in Table 2.9. Thus, the taxonomy being created undergoes structural changes with further semantic analysis input, creating a more complex and complete taxonomical structure. The results, in this example are then structured into a hierarchy such as depicted in Figure 2.11 and with supplemental information added, as shown in Figure 2.12.

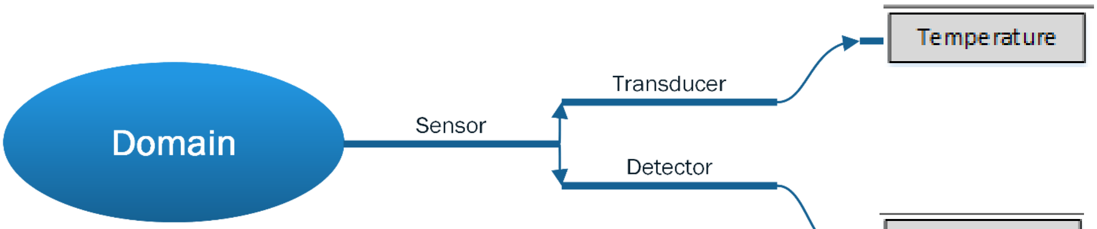


***Figure 2.11 Domain Specific Output***

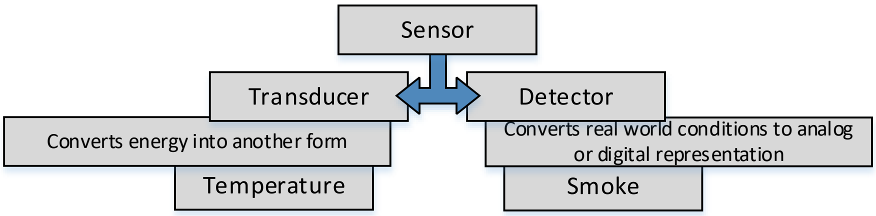

Sensor
Transducer
Detector
Converts energy into another form
Converts real world conditions to analog or digital representation
Temperature
Smoke

***Figure 2.12: Domain specific taxonomy with supplemental information***

#### 2.5.2.2 Semantic Analysis Automation

However, the objective of the second stage of this research, is to provide a method that has the potential to construct a taxonomical structure automatically. This process becomes achievable through the use and adaptation of the enhanced semantic analysis abilities contained within the most recent version of Microsoft SQL. (Note: Section 5.1 reports the failure of MS SQL and the successful substitution of the open-source NLP framework.) The semantic analysis expands the existing full-text search feature in SQL Server, allowing an extension beyond keyword searches. The full-text search integration within Microsoft SQL allows the query of words within documents. The semantic search extension provides the ability to query the meaning of the document, which can be applied to include related content discovery, and hierarchical navigation across similar content. Thus, the automatic search and discovery parameters can be used to identify and establish the document similarity index to identify terms that match a domain description.

This potential is shown in Figure 2.13, which depicts the interaction between the SQL Server Process (SSP), and the Filter Daemon Service (FDS). The FDS provides the textural contents / context through a process of query keyword analysis termed ‘word breaking’.

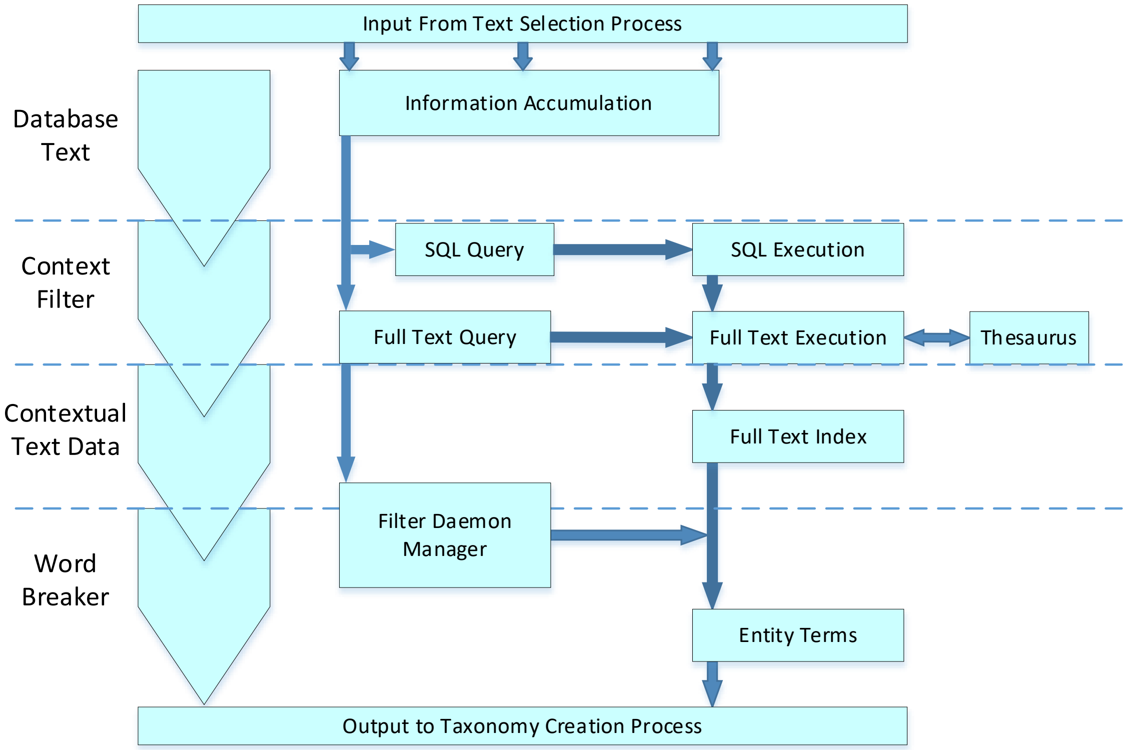


*Figure 2.13: Semantic Engine Process*

The process is further defined to depict the search process along with the filter daemon. The gatherer will parse the indexed documents using information gathered from the word breaker and integrate word list information and remove identified noise words.

## 2.6 CAPABILITY MATURITY MODELING

The Capability Maturity Model was originally developed by the Software Engineering Institute (SEI) at Carnegie Mellon University in Pittsburgh in conjunction with the American Department of Defense. The Capability Maturity Model has been utilized as part of an organizational level assessment with a scale of five process maturity levels (Carnegie Mellon University Product Team, 2002). Each level ranks the organization under review according to the organization's process standardization in the subject area being assessed. The utility of employing this model is that the model can be applied to many subject areas. The subject areas can be diverse and therefore encompasses many seemingly disparate areas as: risk management, personnel management and project management. The SEI Capability Maturity Model was designed to assess software engineering and systems engineering (Paulk, Curtis, Chrissis, & Weber, 1993). The modern applications of the assessment process have been applied to digital investigation and information technology (IT) services (Kerrigan, 2013). The practicality of utilizing the application of the Capability Maturity Model is presented when the maturity model is be used as a benchmark to assess different organizations for equivalent comparison (NIST & Barrett, 2018). It is useful because the SEI Capability Maturity Model can be used to classify and describe the maturity of the organization which in turn can be used to identify the capability of the organization and therefore assess whether the company is capable of completing a project the company is dealing with and then provide the appropriate service to its clients.

The SEI Capability Maturity Model has been changed by the University of Oxford to produce a Cyber-Security Capacity Model, (the Oxford Model) and this has been selected as the basis of the development platform for the Maturity Model artefact output of this research (GCSCC 2014). The reason that the Oxford Model has been selected is that the model has been designed to not only identify and capture a comprehensive and nuanced understanding of cyber capacity, but to also determine ways to enhance the structure and content of a capability maturity model. Thus, the Oxford Model provides a maturity model creation strategy that is applicable to this research model output, because of the structured and logical processes presented. The Oxford model presents a template that provides a visualization of how the factors, aspects and indicators interact in each dimension of a capability maturity model.

This model begins with an investigation into a broad collection of factors from a wide and diverse compilation of information from respected sources. Then there is a pilot phase undertaken, which is used in this research to provide insights into the IoT and cyber forensics as an evolving field of work. The next step involves modification and adaptation of new factors developed through the pilot phase investigation. This process aligns with the design science methodology discussed in Section 3.1 and has also been used when investigating the three case studies addressed in Chapter four. The Oxford model is adopted by this research to develop the Maturity Model artefact through the implementation of the following flow structure shown in Figure 2.13.

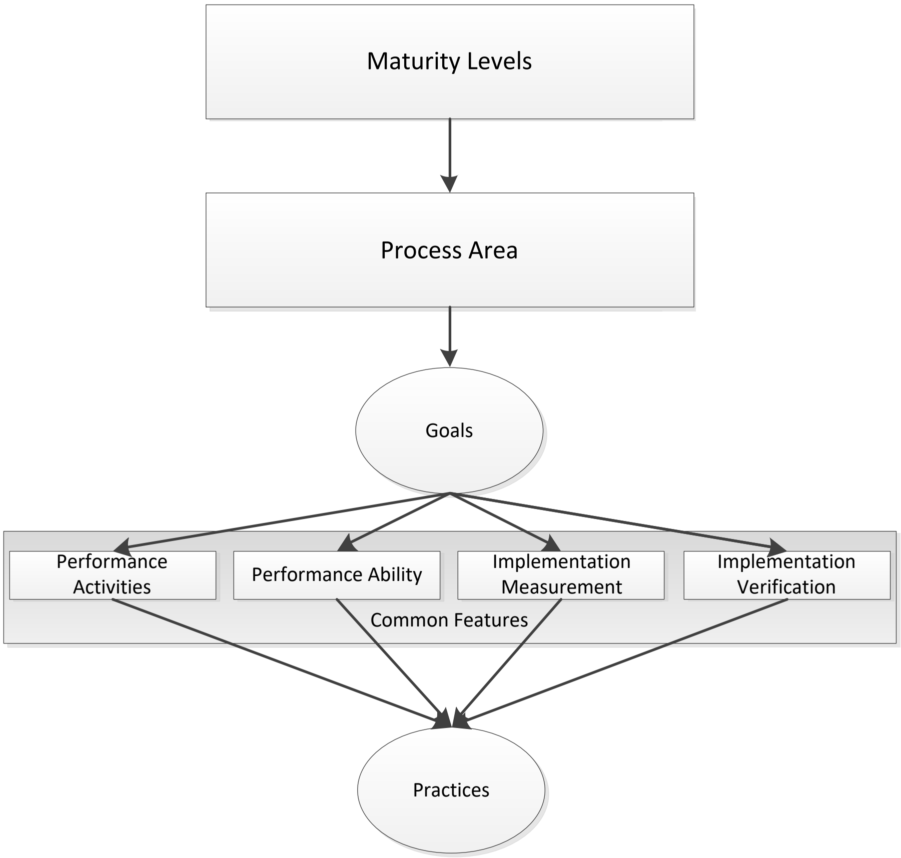


***Figure 2.14: Capability Maturity Model Components adapted from (Carnegie Mellon University Product Team, 2002)***

Figure 2.14 displays the steps that will be undertaken to construct the Capability Maturity Model. The steps begin with establishing definitions of the five levels of maturity, moving on to establishing the Process Area and then the establishment of Goals. Several Common Features that are relevant to the research area are then selected and tabulated, which form a set of practices. Each of the steps shown in Figure 3 are detailed in order.

### 2.6.1 Maturity Levels

Table 2.10 provides preliminary definitions of the five levels of maturity represent a broad, high level perspective and are adapted for the Digital Forensic Capability Maturity Tool for IoT creation. Each maturity level consists of several process areas that have been identified as significant. In order to be considered significant, a set of goals must stabilize a process component. The goal fulfilment occurs upon implementation of key practices. The common features of these practices are grouped into categories, as shown in Table 2.10.

*Table 2.10: Five Levels of Maturity*

| Level | Definition |
|---|---|
| ONE<br>Start-up | Non-existent.<br>Embryonic.<br>Lack of observed evidence. |
| TWO<br>Formative | Formulation of some features of indicators.<br>Ad-Hoc.<br>Poorly designed.<br>Evidence of activity. |
| THREE<br>Established | Sub-factor elements in place.<br>Relative allocation of resources not well considered.<br>Indicator functional and defined. |
| FOUR<br>Strategic | Decisions regarding indicator importance have been taken.<br>Decisions taken contingent on circumstances. |
| FIVE<br>Dynamic | Strategy can be altered, depending on changing circumstances, with clear mechanisms in place.<br>Constant attention to changing environment.<br>Sense and respond.<br>Rapid decision making<br>Reallocation of resources as required. |

### 2.6.2 Process Area

In order to indicate where an improvement focus should take place, each maturity level is decomposed into process areas. The process area classifies the concerns to be addressed to achieve particular maturity levels. To achieve the goals of key process areas requires many different applications depending on the environment or subject area being investigated. It is important that all the goals of the process area are achieved to completely satisfy the process requirements.

### 2.6.3 Goals

The goals are utilized to determine successful implementation of the process area. Thus, the goals can be used to encapsulate the key practices of the process area. The goals are also utilized to define the boundaries and intent and therefore the scope of each process area where the goal attainment will satisfy a key performance attribute.

### 2.6.4 Common Features

Common features are the organization of the methods that describe the key process areas and are attributes used to determine whether the implementation of the key process area is effective. The Oxford Capability Maturity Model identifies five common features, of which four have been selected as appropriate for the development of a digital forensic capability maturity model that is suitable for IoT applications as shown in Table 5 (GCSCC 2014).

***Table 2.11: Selected Common Features.***

| Feature | Attributes |
|---|---|
| Performance Activities | Roles required to implement process area.<br>Procedures<br>Activities<br>Performing, tracking, correcting |
| Performance Ability | Preconditions necessary to implement process to achieve pre-set goals.<br>Resources<br>Capabilities |
| Implementation Measurement | Process is required to be measured.<br>Measurements are required to be analysed.<br>Analysis is used to determine effectiveness of performance activities. |
| Implementation Verification | Verifies that the activities performed comply with established process.<br>Reviews<br>Audits<br>Quality Assurance |

## 2.7 CONCLUSION

Chapter two described the literature search undertaken to establish the context of the research investigation domains: IoT, Cybersecurity and the use of Maturity Modelling for risk identification. The review of literature began with a broad overview, determining relevant context in areas of network security, how network security integrates into the IoT domain, risk and vulnerability. The key contribution ascertained from this portion of the literature investigation identified that determining the risk attributes of the IoT context of an individual device or usage domain presented challenges. Several different IoT models exist, and the relationships between the models are difficult to identify and give conflicting information that is difficult to interpret. Section 2.3 illustrates the challenges by a detailed examination of two IoT communication models, ZigBee and CANbus. The models examined demonstrate the information is disparate, highly technical and requires an in-depth knowledge base, beyond the core business capability of organizations.

Further literature investigation identified a gap exists within published documentation, designed to facilitate understanding between the diverse areas of the IoT domain when attempting to determine risk. Thus, the identified problem is a lack of a standardized presentation of the definition of IoT entities, types of entity, the properties, and types of entity. The key contribution determined the solution required a formal naming, description and itemization process for IoT that produces a model to be used to assist risk enumeration processes. The solution identified through the literature investigation was to provide a Taxonomy. An example of the creation of a taxonomical structure was shown in Section 2.5. However, The Taxonomic output requires adaption, when providing assistance in risk identification for organizational utility. The literature review investigated the Capability Maturity Modelling development and determined that a Maturity Model provided a solution. A Maturity Model creation process provided by Oxford University is adopted to provide an output developed to identify risk within the context of IoT for business use.

Chapter three demonstrates the processes determined from the literature review, to develop the IoT risk identification Maturity Model. The process steps are integrated within the Design Science methodology that gives artefacts and provides expert opinion as an artefact development input at each stage of the process. Each step is outlined, first showing how the Taxonomy will be created by an automated Semantic Analysis process. Then Chapter three shows how the Taxonomy is used to inform the Maturity Model development process. Finally, the testing process of the Maturity Model is defined, and the resultant Risk Identification Maturity Model creation process is described. Each process step is given in depth, showing the links between process steps and showing how the output will inform the next step.

# Chapter 3
# RESEARCH METHODOLOGY

## 3.0 INTRODUCTION

Chapter three defines the research methodology developed and applied for this research. Chapter two investigated the technical aspects of network security architecture, the context of network security for Internet of Things (IoT) and technical risk within the IoT domain. Chapter three provides gap identification and the problem identification from the literature review problem statement contained in Chapter two. The gap identified in Chapter two, is the lack of a comprehensive, yet easily understood, IoT risk identity structure designed for business use. The identified problem, therefore, is that there is a requirement for a taxonomy to be developed, that will assist a business user in the determination of risk attribution. In particular, the evaluation developed in the literature review indicates that the production of a maturity model from the proposed risk attribution taxonomy will provide a solution to the identified problem. Design Science (DS) has been adopted in this research and has been adapted to produce several artefacts that will each inform the next stage of the DS activity application. The DS activities conclude by producing a maturity model as the final instantiation output, where the levels of maturity include risk mitigation capability. The use of case study research is integrated into the research methodology to provide research rigor and validity.

Section 3.1 begins with a taxonomy creation method that is adapted for use in the research, by explaining the practical steps required in a combined process overview. Section 3.2 explains the reasons and justification for the adoption of DS methodology. Section 3.2 gives a logical progression of the DS application process steps that will be undertaken, and the artefacts that will be produced. Section 3.3 describes by order, each of the research phases that the research will use. Section 3.4 establishes the research questions to be answered and the hypotheses that will be tested, and the problem statement for which this research will provide a solution. Section 3.5 investigates the data requirements and provides justification of the use of case studies to provide validation and rigor. Section 3.6 describes the data analysis procedures and evaluation processes undertaken in the research. Finally, Section 3.7 will conclude Chapter three with a brief review of the contents and the outcomes.

## 3.1 TAXONOMY CREATION METHOD

As discussed in detail in Chapter two, the problem identified within the IoT domain is that there is a lack of a simple and easily understood risk attribution processes. The solution to the problem will be provided through the development of an IoT risk maturity model as described in Section 2.9. The risk maturity model development begins with the creation of a taxonomic structure. Section 3.1 describes the taxonomy creation process used in this research.

Section 3.1.1 presents the theoretical basis of taxonomic creation, and describes the semantic processes underpinning the theory. Section 3.1.2 then describes the practical components of the taxonomic creation, as a step-by-step process. Each step, and several sub-processes are defined in depth.

### 3.1.1 Theory

Taxology, for the purposes of this research is used as the process of establishing a common domain language classification structure. The common domain language creation process entails classification identification and naming of terms through a system of vocabulary input and selection processes. As discussed in Section 2.5, the seminal investigations into taxonomical methods relate to the study of living forms, the relationships of living things over time, and the diversification factors exhibited. Taxonomical investigation involves classification derived through the process of naming. However, the concept of systematics, which is the determination of relationships, has been identified by the researcher as being an essential component of this research. Both the concepts of taxonomical creation and systematic determination are therefore utilized within this research and form the basis of the methodological organization structure. The intention is to produce knowledge based, taxonomical output derived from large amounts of data, in the form of text. Thus, the researcher has identified the importance of not only establishing a vocabulary but also identifying possible underlying relationships.

As is discussed in Chapter two, and is further developed in Section 3.2, the Design Science (DS) research method will be utilized to assist the researcher to address the development of new knowledge about the objects under investigation. Of the four research outputs identified as central research outputs of DS: Constructs, models, methods and instantiations, the researcher has ascertained that constructs represent a fundamental core concept to this research. The core concept of the construct artefact type is the definition of a conceptual vocabulary. The conceptual vocabulary, or taxonomy, will be then be used to identify problem domains as part of the model creation process. Models, when describing how things are, describe the relationships between developed constructs, demonstrating the requirement for systematics as an incorporated component of the research methodological process.

The goal of this research, as discussed in Section 2.5, is to investigate a novel taxonomical creation process that will be useful for determining risk identification factors. Also, the evaluation of optimum functionality of the risk identification output from the taxonomy creation process will require evaluation of metrics, the establishment of which is suitable for future research, because the focus of this research is on the assignation of risk identity. As is determined in Section 2.5, a consideration of a 'useful' taxonomy, is that the taxonomical methodology will incorporate several important characteristics. One characteristic is to include a provision that will allow the resultant taxonomy to change and develop over time. The incorporation of change is identified as a requirement for this research, as information systems are in a continual process of evolution and development. An example of the requirement for change, is that additional dimensions will be expected to be added over time. Thus, the taxonomy will be extendible, with the ability to add new characteristics as necessary. Another requirement is that the taxonomical output will be designed to be both comprehensive and inclusive. This requirement means that there should be enough information and dimensions to be relevant within the identified domain and that the process will classify the majority of objects considered within the domain identified. Finally, there is a consideration that the taxonomical output produced will be concise and therefore simple. As the considerations for a comprehensive and inclusive taxonomical output may lead to a complexity caused by the overabundant use of vocabulary, the requirement for a concise output will provide a counterbalance. The effect of the inclusion of these requirement characteristics when designing the taxonomy creation process will provide a taxonomy that is concise and simple, yet comprehensive and inclusive, as well as allowing considerations for flexibility and future change.

As discussed in Section 2.5, a challenge when developing a semantically based taxonomy is the creation of a clear distinction between elements within the taxonomy. The distinction is apparent as a component relationship of individual elements within the field in focus. The irreducible element determination in this instance is an individual unit. The determination becomes problematic, however, when attempting to define an individual or class relationship within the field of risk enumeration, where the relationship can be described as having a higher or lesser risk value comparison. The value comparison output is in contrast to the irreducible output, single element as produced in a botanical taxonomy. Thus, when producing a hierarchal arrangement for a risk identity taxonomy the output is not always complete, or a single value determinant. However, the determination of an individual element as a basic concept, insures exact and specific semantic description for a complete taxonomical determinant. Therefore, the specific description of a risk element, followed by establishing a distinct and individual semantic determination, is the focus of this research.

Hence, the research focuses on the establishment and development of assertions, where a precise semantic relationship is established between the subject and the assertion about the subject. The establishment of the precise relationship limits the assertion differentiation to different kinds of meaning, rather than a provision of all variants of semantic possibilities. This reduces complexity that can be produced through investigation of different classes of nouns or verbs instead of the relational assertions. Thus, the assertion determinations investigated within this research develop through stages, from Domain, through Identity, to Characteristic, and finally toward an Attribute assertion determination.

### 3.1.2 Practice

The practical component of the taxonomy creation process within this research involves a systematic progression through several steps. Section 3.1.2 presents each of the steps to be undertaken and describes the significant points within each step. As discussed in Section 2.7, the purpose of the taxonomy that is created when following these steps is to extract the relationships among factors, which then may be used to identify factor relationships. The practical components of the taxonomy creation follow four broad steps: topic identification, link establishment, information vocabulary establishment and taxonomy population. The four steps, which are shown in Figure 3.1, contain several internal, intermediate steps, for completion.

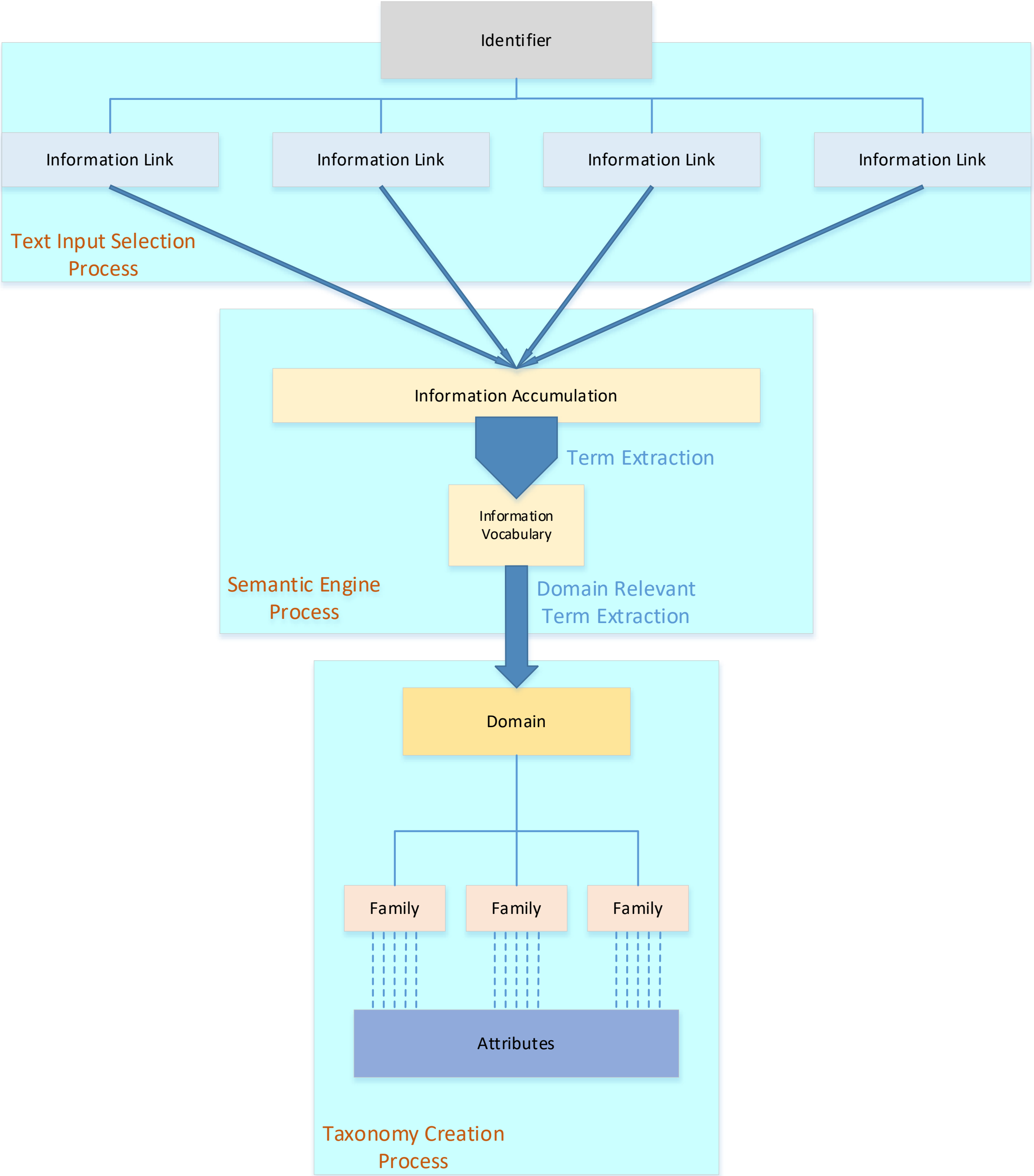


***Figure 3.1: Combined Process Overview***

The process begins by organizing a large body of general information around a top-level concept which is used to establish the common domain classification structure. The top-level topic focus is labelled the identifier, as shown in Figure 3.1. The second step is to establish links to the specific information data that will be input into the creation process. The information links will be informed by information derived from the topic identification process performed in step one. The links provide large amounts of data, in the form of documents. The documents are reduced to text, which is de-structured and analyzed. The analysis results will then provide the input into the third step of the taxonomy creation process. The third step is to establish the information vocabulary, which is determined by ascertaining relationships between individual components of the text data input from step two of the process. The fourth and final step of the taxonomy creation process utilizes the relationship information derived as an output of the third stage to populate the completed taxonomy structure. Each of the internal processes of the four steps are outlined in detail in Figure 3.2 and the following paragraphs.

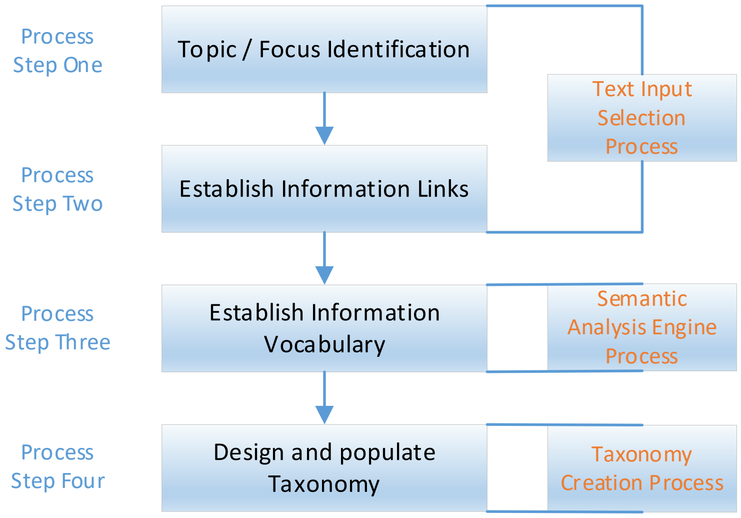

Process Step One
Topic / Focus Identification
Text Input Selection Process
Process Step Two
Establish Information Links
Process Step Three
Establish Information Vocabulary
Semantic Analysis Engine Process
Process Step Four
Design and populate Taxonomy
Taxonomy Creation Process

***Figure 3.2: Practical taxonomy creation steps***

Process Step One: Topic / Focus Identification: The Taxonomy creation process begins with the selection of a term that establishes the top-level taxonomy classification identifier, as shown in Figure 3.2. The identifier is determined by assignation of a broad field or topic. The purpose of the identifier determination is to provide an overarching concept that is used to establish the common domain of the taxonomical investigation. The identifier is used to establish the scope and extent of the information that is used as links to the data input for the taxonomy creation. The identifier provides limitation direction for the data input and stipulates a primary categorization factor. Therefore, the identifier is used to ascertain which journals, best practice category or standards that will be input into the taxonomy creation process.

Process Step Two: Establish Information Links: The information links are determined from the output of step one, which establishes the overarching domain of the taxonomical investigation process. The links, as detailed in Section 2.1 establish the characteristics of comprehensive yet concise data input for a useful taxonomy. The information links provide a structured input which is utilized to select the data sources of information. The sources of information are (see Section 2.1): Peer reviewed journals, best practice recommendations, and internationally accepted standards. However, without determining scope and direction by establishing information links, the amount of data is too voluminous to be reliably processed. Therefore, the links to the information to be parsed in step three of the taxonomy creation process provide a reduced dataset. As shown in Figure 3.3, step one and step two of the taxonomy creation process are encapsulated as the text input selection component of the combined process. Thus, the links provide data output, that is then input as accumulated information in the form of text, to the third step of the taxonomy creation process.

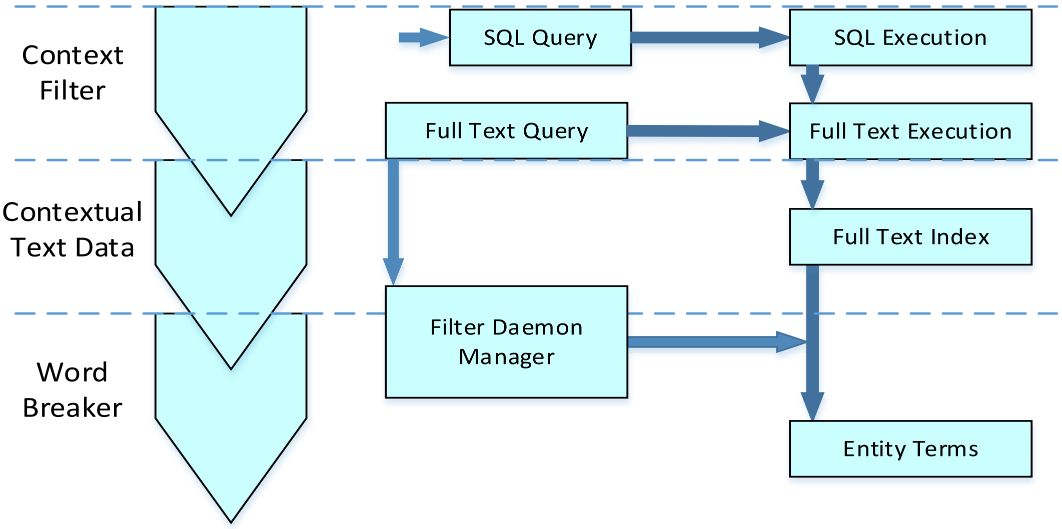


*Figure 3.3: Semantic Analysis Engine sub-process*

Step Three: Establish Information Vocabulary: In step three, the establishment of the information vocabulary contains several internal process steps, as shown in Figure 3.3. The third step is complex and multifaceted and is presented as the Semantic Analysis Engine (SAE) process. The SAE process takes the input of accumulated information, derived from the second step in the taxonomy creation process, and through the application of the internal steps outlined, extracts domain relevant terms. The three internal, sub- process steps of the SAE begin with a context filter activity and then a second sub-process activity of contextual term extraction is performed, and the results are output into an information vocabulary consisting of a full text index. An interpretation activity of the full text index information vocabulary is then performed as the third sub-process, presenting an output of domain relevant entity terms, which then inform the taxonomy population process.

SAE Sub-Process One: Context Filter: The association rules pertaining to significant relationships are mined, or extracted from the large, but specific datasets, of accumulated information. A five-step process of data mining is outlined in Section 2.1: Selection, Pre-processing, Transforming, Data modelling and finally, Interpretation. The selection and pre-processing components of the data mining process steps are undertaken in the identifier and information link steps one and two. Therefore, step three of the taxonomy creation process is data transformation and data modeling. The SAE inputs the accumulated data in the form of text, and applies a dimension reduction process, presenting an output of transformed and conditioned data. This means that the data is inspected, and non-relevant data is rejected from the dataset. Relevant text data is determined by evaluation against a thesaurus of domain relevant terms, as defined in Section 2.5.1.2. The output from this process step results in a smaller, manageable text dataset, producing the input for the second sub-process, the information vocabulary establishment.

SAE Sub-Process Two: Contextual Text Data: The conditioned text data output from the first sub-process presents an input to the second sub-process in the form of semi-structured text data, as shown in Figure 3.3. The second sub-process activity is designed to further filter the data which provides a fully structured output into the filter daemon. The contextual text data is formed through the application of a filter process, which takes the output of the full text execution from the context filter which has been compared to a thesaurus of domain relevant terms. The text data input to the second sub-process is then subjected to a feature filter as the top layer of the filter daemon manager. The second sub-process will present a full text index as a concurrent output, through a deep indexing process. The fully indexed text data will in turn be utilized in an evaluation process with the output of the filter daemon in the third sub-process activity.

SAE Sub-Process Three: Word Breaker / Text Mining: Keyword term extraction, as the third SAE sub-process activity is incorporated into the filter daemon manager, which is shown in Figure 3.3. The lower layers of the filter daemon consist of four components: clustering analysis, modeling, term graphing, and the application of association rules. Each of these components are used by the SAE to perform automated patterning analysis of the conditioned text dataset output by sub-process one. Clustering provides an automated process of unsupervised text categorization to the SAE. Modelling provides a second automated process that inputs the identifier term output derived in the first topic focus step and extracts the matching topic clusters. The topic clusters are then rated by performing an automated term graphing process which evaluates the frequency and proximity of topic terms within the matching topic clusters. The final process within the SAE text mining activity is to apply association rules. The association rules provide the SAE relationship discovery processes using the output from the previous term graphing activity. The association rule set analyses support count, occurrence frequency and distance to provide indications of relationship strength. The relationship strength is determined by rules of association that satisfy minimum confidence thresholds and minimum support threshold. Each of the four components are therefore utilized by the SAE to identify relationships between terms and the domain specific topic identifier. The relationship identification process is used by the SAE to retrieve domain relevant, structured keyword terms in the form of entity values. The resultant entity terms are then output to the fourth and final stage of the taxonomy creation process, design and populate the taxonomy structure.

Step Four: Design and Populate Taxonomy: The activities carried out in step four of the taxonomy creation process present an output of a taxonomical entity classification organization. The output derived from the previous steps are input into the fourth and final step, populating the taxonomy entity structure as shown in Figure 3.4. The identifier is determined as the top-level concept and focus. The identifier is established as the output of process step one and provides the input as the overarching taxonomic classification. The domain and family entity values are determined as the output from the context and the contextual text data sub-process steps of the Semantic Analysis Engine (SAE) at the upper layer of the filter daemon. The relationship information provided by the third sub-process of the SAE, when compared to the full text indexed output of the context filter provide context structured attributes. The attributes are filtered by domain at a high-level classification by the context filter component, detailed in sub-process one, providing domain context. The attributes are then subject to the second filter activity of sub-process two, providing family context. The lower filter manager activities of sub-process three provide relational entity terms that populate the lower level, attribute entities of the taxonomy structure.

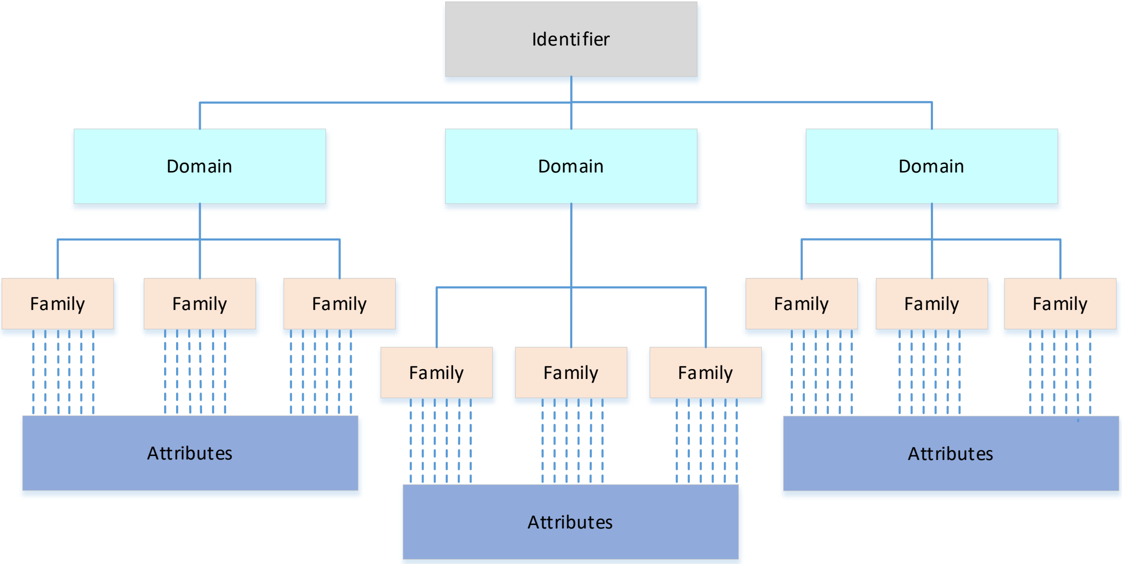


*Figure 3.4: Taxonomy Entity Structure*

## 3.2 RESEARCH DESIGN

Using a Design Science (DS) method investigates how adding to existing knowledge will help resolve an identified problem. There must be, therefore, an identifiable gap in the existing knowledge base that has been discerned by the researcher before seeking to adopt a DS process. There is an additional requirement, which the researcher seeks to develop and communicate the findings, in terms of resultant additions to the knowledge base concerning both the management of information technology and the use of information technology for managerial and organizational purpose (Hevner, March, Park, & Ram, 2004). Adding information in the form of new knowledge through the development of artefacts can be a complex process and designing useful artefacts is difficult due to the need for original advances in domain areas in which existing theory is often inadequate. Adding new knowledge within the domain of Information Systems (IS) requires two distinct but complementary models, behavioral and design science whereby behavioral science is based upon natural science, and design science is founded upon engineering (Alturki, Gable, & Bandara, 2013).

Technology and behavior are not mutually exclusive in IS, they are in fact interrelated (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). Thus, the adaptation of DS for IS research methods presented within Chapter three have been evaluated with three case study demonstrations, as reported in Chapter four. The contemporary phenomena investigated within this research, are security aspects within the IoT domain. The contextual domain investigated within this research is in the context of business IS capability. Therefore, the phenomenon of security within the IoT and context boundaries of business IS, may not be immediately apparent when evaluating risk attribute assignation (Cusack & Ward, 2018). The inclusion of the case study evaluation process will provide rigor to the research, as demonstrated in Chapter four by allowing analysis on multiple levels and provides understanding of how both the phenomenon and context are interrelated. Thus, the use of case studies in research, tests the validity of the output of the DS process within a real-life context.

### 3.2.1 Design Science Research Application

The application of the Design Science (DS) research processes in this research is intended to provide the output in terms of artefacts. The application of the DS processes provides a research focus upon the development and design of output artefacts, as discussed in Section 2.1. Thus, throughout the artefact creation process, knowledge is gained, and the dissemination and publication of the resultant knowledge is inherent within the DS research process. As shown in Figure 3.5, the DS research processes have been adapted for application within this research, where the DS process sequence is listed on the left and the process sequence for this research is listed on the right. Each of the process sequences, or activities are now described.

The first activity within the DS process utilised in this research is initialised with an artefact output that is designed to define the specific research problem investigated. As discussed in Section 1.2, the identification and conceptualisation of the research problem is important, because the identified problem complexity provides justification of the value for an effective solution in the form of an artefact. The definition of the value of the solution provides understanding of the reasoning underpinning the researcher's designation of the problem's level of importance. The value of the solution also determines the researcher's motivation to present the solution. Therefore, the first activity provides an inference drawn from the literature review evidence and reasoning and links the first DS activity with the second.

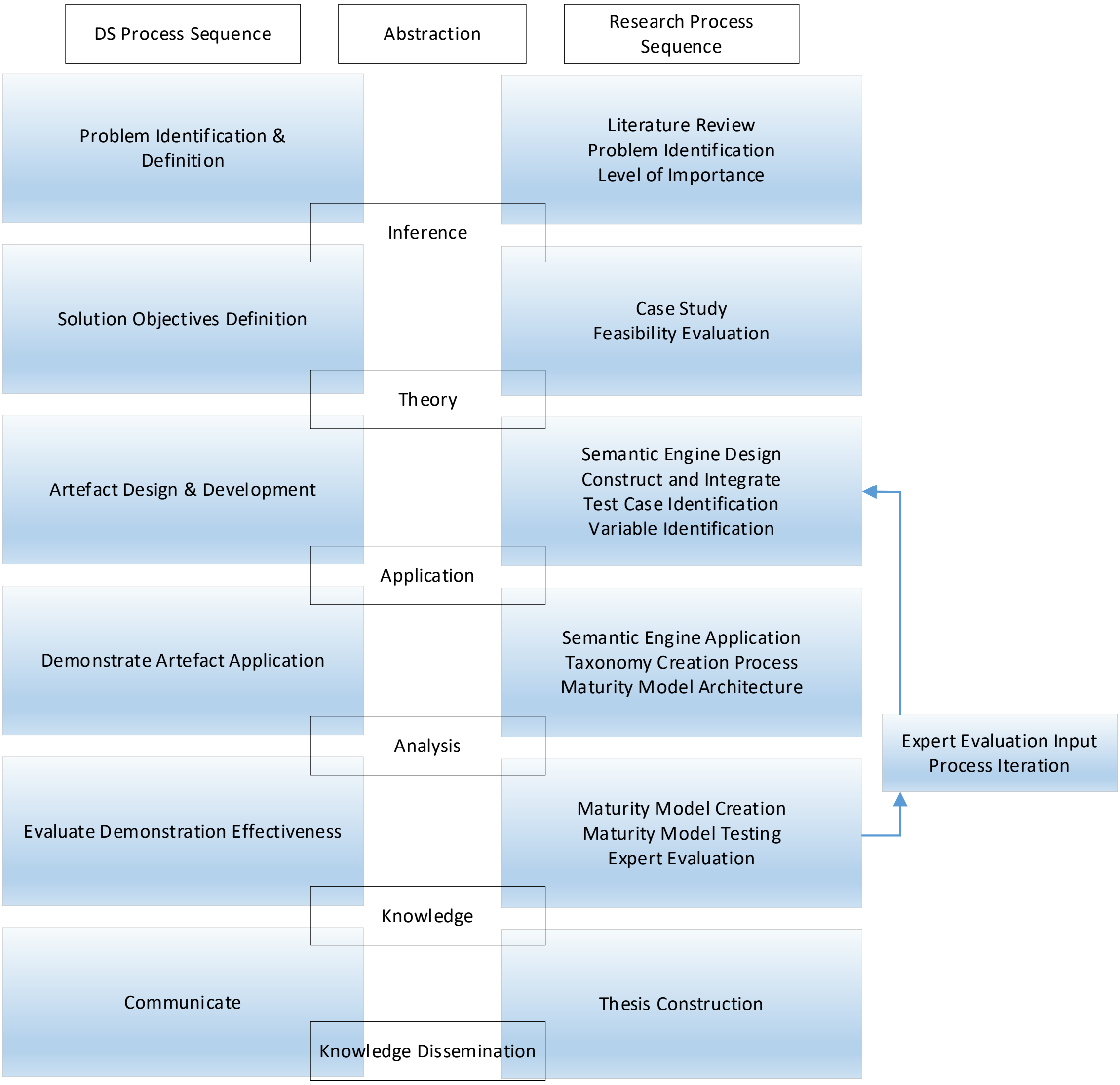


***Figure 3.5: Design science research application overview***

The second DS activity is the output from the problem definition. The objectives for an effective solution are determined, in the form of inferences, from the problem definition output of the first activity. The second activity requires an input of knowledge of what is possible, and for this research, that which is feasible. The requirement for an effective application of the second activity is an input of knowledge about the state of problems and the efficacy of current solutions, if any. The objective of the second activity step of the DS process, is to theorize objectives in which a desirable is better than current solutions, or to define a theoretical artefact that will support novel solutions to the problem definition. For the purposes of this research, three case studies are evaluated, to provide a feasibility check, and validation of the steps from inference to theory. The output from the three case study evaluations, are presented in Chapter four. The theory output of the second activity gives the input to the third DS process sequence, where the theory input is processed into an application output.

The third activity of the DS process is to design and develop the proposed artefact theorised in the second activity. The contribution of the research is embedded in the design of the artefact output of the third activity. As discussed in Section 3.2.2, the design of the various artefacts is determined by identifying the functionality and process requirements of a potential solution to the research problem. The third activity, therefore, is directed to designing and creating an effective artefact. The functionality and process requirements for the proposed artefact are determined from the theory output of the second activity. The third DS process phase is multifaceted, and therefore complex. Artefacts consist of quantifiable outputs of several artefact types that may be considered both tangible, when considering instantiations, and intangible, when considering methods and models. However, all artefact outputs are designed to assist researchers and practitioners when analyzing and addressing a solution to the identified problem defined as the first activity output. Developing systems that successfully implement IS artefacts in an organizational context may require behavioral science research validation. Validation is required to explain the phenomena with respect to the artefact's use, usefulness, and impact on individuals as well as the organization, quality and dependency of the organization on the system (Delone & McLean, 2003). The knowledge contribution of this research is enveloped within the third activity of the DS process and comprises the design and development of several different artefact types. An expanded explanation of the development and design of the artefacts produced from this research are described in Section 3.2.2 of this chapter. Upon the completion of the third activity of the DS process, the output in terms of artefacts is then able to be applied and tested for effectiveness of solution for the identified problem.

The fourth activity of the DS process begins with demonstration of the use and application of the designed artefact output from the third DS process activity. The demonstration of the artefacts begins with proof that the artefact works. The proof will begin with a binary determination of either: Yes, there is an output of risk attribution, or No, there is no output. Once there is a determination of risk attribution output, the fourth activity will proceed towards a refined, graduated output that will be utilised as the input to the fifth stage of the DS process. As discussed in Section 3.5 and Section 3.6, the selected demonstration involves the use of case studies and technical experimentation to provide initial Proof-of-Concept of the application of the artefact usage through knowledge to solve the identified problem. Therefore, the fourth activity will involve an internal development process that will mature the artefact design so the output can be used to inform the creation of the maturity model, as part of the fifth DS activity.

The fifth activity of the DS process takes the output from the previous activity, to provide the input of the demonstration effectiveness evaluation process through expert analysis. The evaluation of effectiveness is problematic, as relevant metrics and analysis techniques may be difficult to establish. Therefore, as discussed in Section 2.6, the utilisation of the output of the demonstration phase to inform the development of a Maturity Model (MM) has been selected as the evaluation component of the fifth activity. The MM will then be provided for expert evaluation. The expert evaluation of effectiveness will provide input for artefact refinement through an iterative activity input to DS phase 3, artefact design and build activity. Therefore, the expert input will provide observation and measurement quantification in terms of how well the artefact output supports the solution to the identified problem. The output from the iteration of this activity will provide information in the form of knowledge that is novel and enduring. The knowledge will then be disseminated through the release of the research output in the form of a PhD Thesis, as an integral component of the sixth and final DS process activity.

The sixth activity of the DS process for this research is the construction and publication of a PhD thesis. The motivation and purpose of the final activity is to communicate the importance of the identified problem and the novelty, utility and endurance of the artefact as a solution to the problem. As an integral element of the research output structure, the rigor of the design and the validation of the output shall be demonstrated. The effectiveness of the design and the use of the findings as a basis for future research shall be identified. The contribution to IS will be identified and the specific nature of these contributions discussed as part of the knowledge dissemination output.

### 3.2.2 Design Science Artefact Output

Design Science (DS) research processes provide outputs that are intended to deliver additions to Information Systems (IS) knowledge through the application of the six design phases, identified as activities and described in Section 3.2.1. Activity three described in Section 3.2.1 consists of an artefact design and development process. Activity one and two identify a problem and itemize the objectives for a solution to the identified problem. The activity of determining functionality and the process of developing the architecture is central to the creation of an artefact. Thus, the inception, design and development of DS artefacts for IS is complex. Therefore, this section describes the artefacts that will be developed as part of this research and the relationship between the artefacts as part of the DS processes.

Artefacts as defined by (Nunamaker Jr, Chen, & Purdin, 1990; Vaishnavi & Kuechler, 2015) consist of several different types, which can be defined broadly in the following manner:

- Inferences, which consist of conclusions derived from background knowledge and contextual clues based upon evidence and reasoning. An inference artefact is utilized within this research to provide the investigation domain which gives research scope and forms the basis of the literature review. The inference artefact is an output of the case study evaluation process of the research. Adding an inference artefact to the list of artefact definitions is discussed in Section 2.1 and is a novel contribution to DS for IS produced from this research. The outcomes that are developed from a semantic analysis of the inference artefacts form the basis of the vocabulary for the construct artefact.
- Constructs, consisting of vocabulary and conceptualizations, are utilized to accurately describe the identified problem, provide components, and objectives for the solution. The construct artefact provides the conceptual terms of reference for the problem and solution domain that evolves and is refined throughout the design application phases.
- Models, which consist of abstractions and representations are utilized to symbolize a problem and the associated solution domain. The model artefact provides a set of propositions or statements that can express relationships among constructs. Thus, the model artefact is a development of the construct artefact, where the model focuses upon utility rather than conceptualization.
- Methods, consist of algorithms and process guidelines, which are used to perform a task within the solution domain. The method artefact is utilized to manipulate the construct artefact to develop the solution statement. Thus, the method artefact can be used to design the instantiation artefact.
- Instantiations, which consist of prototypes and system implementations, represent the research outcomes. The outcomes are developed from an analysis of the application of the method artefact.

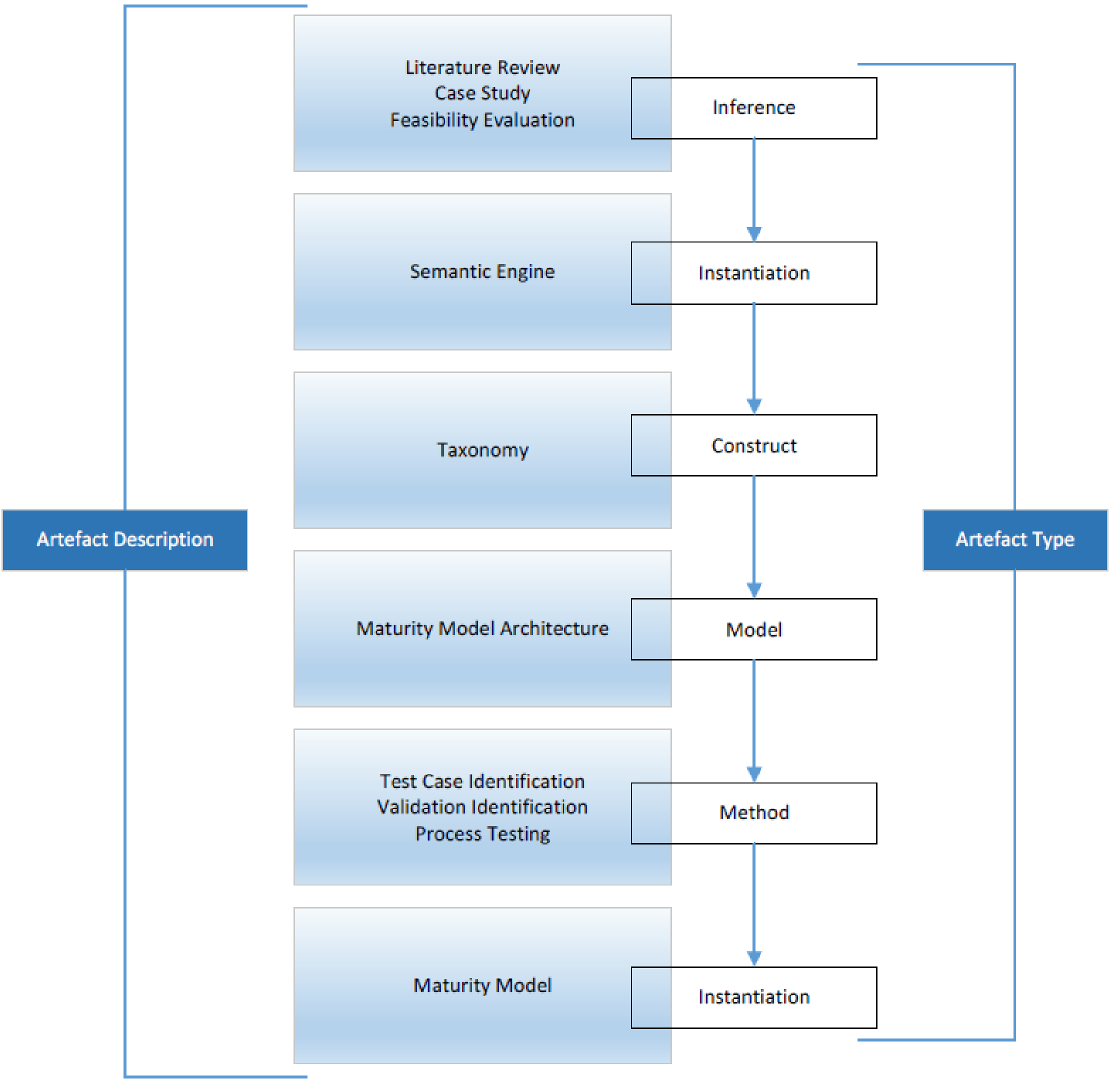


***Figure 3.6: Artefact types created.***

There are five types of artefacts created as part of the Design Science (DS) process implemented as part of this research. The artefacts described are created at various stages of the DS implementation and are used to inform the next stage of the DS process, as shown in Figure 3.6. The six artefacts are output from this research, with the novel research contribution of an inference artefact.

The initial artefact design produced as part of the DS application is an inference artefact. The inference artefact presented in this research is the product of conclusions drawn from the literature review presented in Chapter two, followed by a validation process through the case study feasibility evaluation reported in Chapter four. The inference artefact provides the scope and overarching domain identification for the research scope. The inference artefact identifies the paradigm as risk enumeration, within the Information System domain. The inference, in terms of an identified problem definition comes from the output of the literature review knowledge gap identification.

The inference drawn is tested with a real-world application of a case study providing a conclusion based upon evidence provided by the case study analysis and reasoning derived from the researcher’s background knowledge. The problem identification inference artefact is therefore the combined output of the first two DS activity processes and provides the foundation for the artefacts that follow.

The second artefact produced, as shown in Figure 3.6 is an instantiation artefact. The design and construction of the Semantic Engine is the initial instantiation artefact and is the third stage application output derived from the input from the research outcome of the first and second activity process of the DS application. The Semantic Engine, as discussed in Section 2.5 is a system implementation prototype designed to provide the input, or inform the Taxonomy generation process, as shown in Figure 3.6.

The third artefact output is a construct in the form of the populated Taxonomy. The Taxonomy is a vocabulary based; symbolic representation of the problem solution domain determined by the inference artefact. The Taxonomy is subjected to an evolutionary process of development through refinement of the Semantic engine instantiation artefact. The Taxonomy construct creation process provides output in the form of conceptual terms of problem solution relationships, designed to inform the Maturity Model architecture.

The fourth artefact output is a model, in the form of an abstraction which represents a Maturity architecture. The Maturity architecture model consists of a series of statements, which will determine relationships based upon the input from the taxonomy construct. The Maturity model artefact focuses on business utility and provides the basis of expert opinion generated through testing of the Maturity model. The testing of the Maturity model forms an integral component of the fifth artefact.

The testing and validation activities produce the fifth, or Test method artefact output, in the form of a process. The design of the Test method integrates business case studies with the goal of manipulating the Taxonomy construct and analyzing change, if any. This Testing method or process is designed to provide validation of the identified problem solution. The output is evaluated by expert opinion, and a determination formed on whether a change iteration is required. If there is a change iteration requirement, input changes to the Semantic engine, as part of DS activity phase three may be required. After the expert evaluation input has been integrated into a refined DS process, a final Maturity model output is produced.

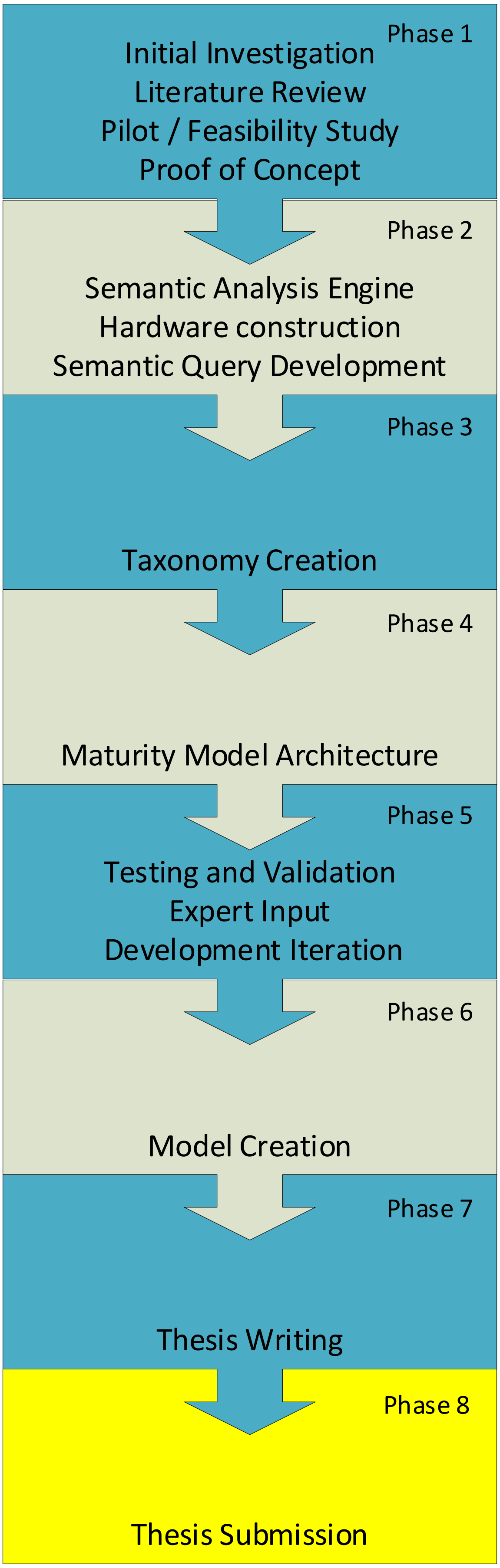


***Figure 3.7: Research Phase Diagram***

The sixth and final artefact output is an instantiation of the Maturity level identifier. Thus, the Maturity model moves from an abstraction representation towards an implementation of a prototype system. The Maturity model in the form of a Maturity level identifier instantiation represents the outcome result. The Maturity model instantiation provides the solution to the identified problem addressed by the inference artefact.

Therefore, the proposed research will be developed to assess a specific aspect of the Cyber domain, the Internet of Things, and specifically the investigation into assignment of risk levels. From a business viewpoint, the Internet of Things domain embraces many aspects of business operations are difficult to assign risk (Atzori, Iera, & Morabito, 2010). The use of a tool that can establish risk through investigation of Maturity levels can be used to fulfill the task of risk identification, evaluation and therefore identify areas that will improve Capability Maturity.

## 3.3 RESEARCH PHASES

Each of the proposed research phases will integrate key aspects of the DS processes outlined in Section 3.2. Each of the proposed phases are listed in Figure 3.7, whereby the objectives, design and development of each artefact, whether a construct, model, method or instantiation are evaluated by selected experts. The initial feedback from the experts will used to validate the proposed solution's objectives, and each subsequent evaluation iteration provides information to be fed back into the design and development stage of each phase.

### 3.3.1 Phase One: Initial Investigation and Proof-of-Concept

Phase one of this research consists of an investigation into the current state of risk enumeration for the Internet of Things. The preparation for the investigation consists of a systematic literature review that encompasses a wide range of disciplines and domains. Upon the researcher consolidating the information contained within the literature examined, the initial research identifies a gap in the current state of knowledge. As shown in Chapter two, the researcher found that a Maturity Model can be developed that assists businesses to identify risk and risk mitigation. Maturity Model development processes were investigated, and Maturity Model Components were adapted to provide a basis for the research. The analysis of the process required to develop the Maturity Model is the expected output of this research and will contain the contribution from the research.

Section 3.5 discusses the requirements for validation and rigor. The researcher uses a case study comparative analysis method to provide a Proof-of-Concept and feasibility study of the assertions formed from the literature review process. Three Case Studies were selected for comparative analysis, because they represent widely disparate scenarios of structure, attack vector and actor motivation. Thus, the risk identification, as the output from the test case scenarios, validate the proposed processes and methods, and shape a risk taxonomy.

Adapting DS research models as outlined in Section 3.2, have been incorporated into this research phase. The specific research problem has been identified and defined through this research phase, and the value of the proposed solution has been justified. The identified problem definition has been used to propose an artefact as a potential solution. The problem has been analyzed conceptually indicating that the proposed Maturity Model (MM) captures the identified problem's complexity. The objectives of the proposed MM inferred from the literature review process and defining the problem provides a potential solution. The objective is therefore, to develop a model that is better than the current Maturity Models. The MM proposed as the outcome of this research addresses identified gaps in the current models.

As shown in Section 2.6 the process of creating a MM, has overarching requirements for the creation of a taxonomy structure. Therefore, the creation of a taxonomy from the manual evaluation is the artefact from the test case evaluation. The taxonomy artefact produced by the pilot study, is a broadly defined artificial construct that provides an instantiation. The taxonomy output from the pilot study will be developed further through each of the subsequent research phases and utilized to produce the final research output of a risk enumeration MM. The initial investigation activity determined the taxonomy creation process, its desired functionality, potential architecture and integration into the following research phases.

Section 3.2.1 defines a DS requirement, where the researcher communicates findings relating to the artefact output to be evaluated for improvement by experts. The evaluation process forms an improvement design loop, where the expert opinion is input into an earlier activity of the DS process. The phase one artefact output of the completed case study feasibility evaluation is demonstrated to experts. The demonstration is designed to address the research phases, inform the semantic engine development, and assist the potential risk MM construction. The proposed MM artefact is evaluated by the experts, and also through an internal expert, Post-Graduate Research Faculty review process. The feedback received for each of the expert bodies has been integrated as an iterative step into the steps of re-evaluating the CMM's objectives and to improve the proposed model's effectiveness. The evaluation process that incorporates expert feedback as an iterative improvement cycle is a major part of each application phase of the DS process.

### 3.3.2 Phase Two: Semantic Engine Development

Phase two of this research develops a Semantic Analysis Engine (SAE). The SAE will be designed to provide automated taxonomical information. Information regarding Best Practices and Regulatory information is entered into the Semantic Database, and peer reviewed journal papers. The Semantic information forms the basis for the constraint definition. The engine provides essential elements for the testing process and thus is an essential component of the development of the Tool. Indications of risk elements are identified and assigned an initial weighting.

The SAE's desired functionality is identified. It has been determined that a minimum hardware architecture is required to provide a platform for the semantic engine development. The SAE requires an upgraded hardware platform that can be integrated and built into a dedicated server. A Microsoft SQL Server version of the latest build, 2017; RTM 14.1709.3807.1, is installed. The platform provides the core Semantic Engine development structure. The internal steps of phase two is installing, constructing and testing the hardware platform, and then developing the software analysis capabilities of the semantic engine.

The indication that this phase is complete is when the Semantic Analysis Engine can output a basic taxonomy structure from a process of semantic analysis of the input, as shown in Figure 3.2. When this objective has been reached, the Semantic Engine will be established as a Design Science artefact, in the form of an instantiation. The Semantic Engine, as an instantiation artefact, informs the taxonomy generation process as discussed in phase three of the research process.

### 3.3.3 Phase Three: Taxonomy Creation Process

Phase three of this research develops the taxonomy structure creation process as described in detail in Section 3.1. The taxonomy creation process of phase three utilizes the output from the Semantic Analysis Engine to populate the taxonomy structure design of identifier, domain, family and attributes as shown in Figure 3.4. The objective of phase three is to automate the taxonomy creation process to provide a Design Science artefact in the form of a construct. The taxonomy construct will be analyzed for utility by comparing three test case outcomes as defined in Section 3.5 and manually evaluated in Chapter four. The desired outcomes are that the taxonomy construct is adaptable to change, comprehensive, and yet concise. The SAE process will be adjusted for each of these required outcomes, to produce an output that will improve the taxonomy output of phase three. The adjustment process will be in the form of a DS development iteration loop, between the taxonomy and the SAE process steps. The taxonomy construct artefact as the final output of phase three will be used to inform the fourth phase maturity model architecture design.

### 3.3.4 Phase Four: Maturity Model Architecture Design

The taxonomy artefact construct output from research phase three is utilized to provide Maturity Model (MM) architecture components. The architecture defines each of the components of the MM model. The components are used to develop, extend and enhance existing frameworks which are used in the testing process to develop the proposed artefact in the form of a model. An extensive investigation of literature defined MM development processes. As discussed in Section 2.6, there are many different MM models available, but very few definitions regarding MM model creation processes. Therefore, the Oxford Model has been adopted to provide the MM creation design adopted in this research. Thus, as discussed in Section 2.6.2 the MM components output from research phase four are designed to conform to the following terms as defined in the Oxford model: Dimension, Factor, Category, and Indicator. As shown in Section 2.6, the Oxford MM construction framework begins at an overarching top level within a MM taxonomic structure, identified as a Capacity. The Capacity entity consists of Dimension entities, which in turn consist of Factor entities, as identified by the Oxford model. The Oxford taxonomic entity structure is incorporated into the MM creation process utilized in this research. The labels of each entity are changed for the taxonomic output of this research. The entity label change is undertaken to transfer a focus from the capability modelling toward IoT risk identification modelling. Therefore, shown in Figure 3.8, the research model IoT risk identification taxonomic creation process is aligned to the Oxford model in entity structure but utilizes different entity descriptions and terms (GCSCC 2014).

As discussed in Section 2.6, a traditional capability MM provides benefits to the target of the MM, when the model is utilized as an improvement framework. The MM improvement frameworks investigated in Section 2.6 identify the business organization as the target of the model, and therefore, the benefit of the MM is the provision of organizational improvement. However, as distinguished during phase one this research is using risk attribution as the target for the MM improvement framework. The focus of this research is to provide an attribute output in the form of the MM framework designed to provide IoT risk identification and therefore indicate risk improvement strategies to projects, teams or individuals. Thus, the MM framework output from this research is to provide risk identification that is close to the application of the device layer, (see Appendix A). The process of risk identification, however, consequently supports enterprise level improvement objectives and strategies. Therefore, the Oxford maturity model key terms are aligned to the research phase three taxonomy output entity identification when creating the research Maturity Model, as shown in Figure 3.8 (GCSCC 2014).

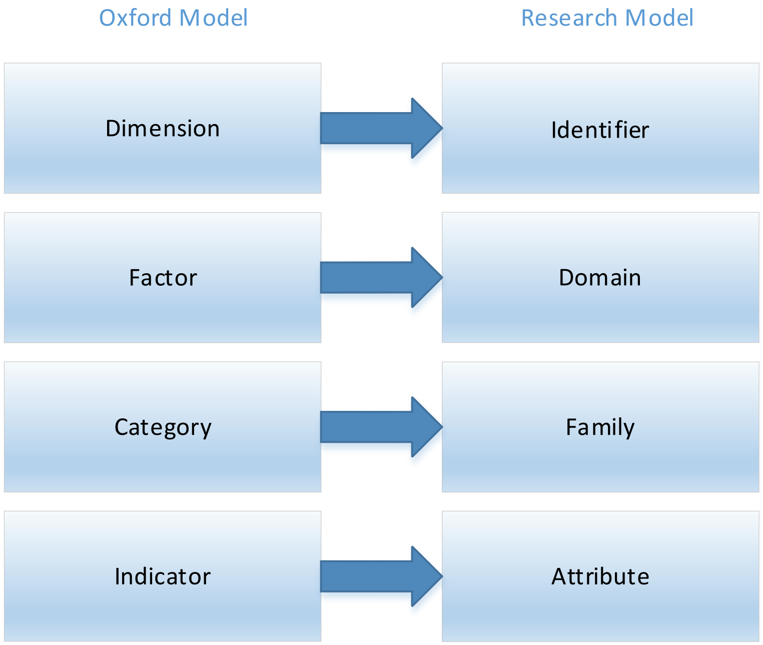


***Figure 3.8 Oxford MM and research MM term alignment***

The objectives of this as a CMM artefact will be shown and demonstrated to the selected experts. Feedback will be sought and integrated as an iterative process to improve the MM's creation processes as shown in Figure 3.8. The iterative demonstration and feedback integration process is expected to occur at least twice.

### 3.3.5 Phase Five: Testing and Validation.

A number of test cases are identified as suitable for testing the MM artefact output from research phase four. The test cases will involve various aspects of IoT vulnerability exploitation. The testing process provides hardware / technological aspects to be investigated as part of the proposed validation testing method. The validation testing method provides a DS output in the form of a method artefact. The artefact method for validation testing, inputs information from the case studies to ascertain the validity of the MM architecture as described in Section 3.5. Each case study provides text data information that furnishes input into the SAE and then is subjected to a manual analysis as separate parts of the testing process. The testing process will integrate taxonomic information taken from the semantic analysis produced by the semantic engine and will then be compared with the manual output. The process will provide adjustment capabilities to the SAE and therefore taxonomy output. Expert opinion will be sought as part of the Design Science development iteration loop.

The Case Study analysis provides data that will be used to setup a hardware testing test-bench. A single variable will be selected at a time, and the attributes manipulated. The changes in the dependent variables are then fed back into the Semantic Analysis engine so that the process will be enhanced with each testing iteration. The final output is a model for assessing the Maturity Model as shown in Figure 3.9.

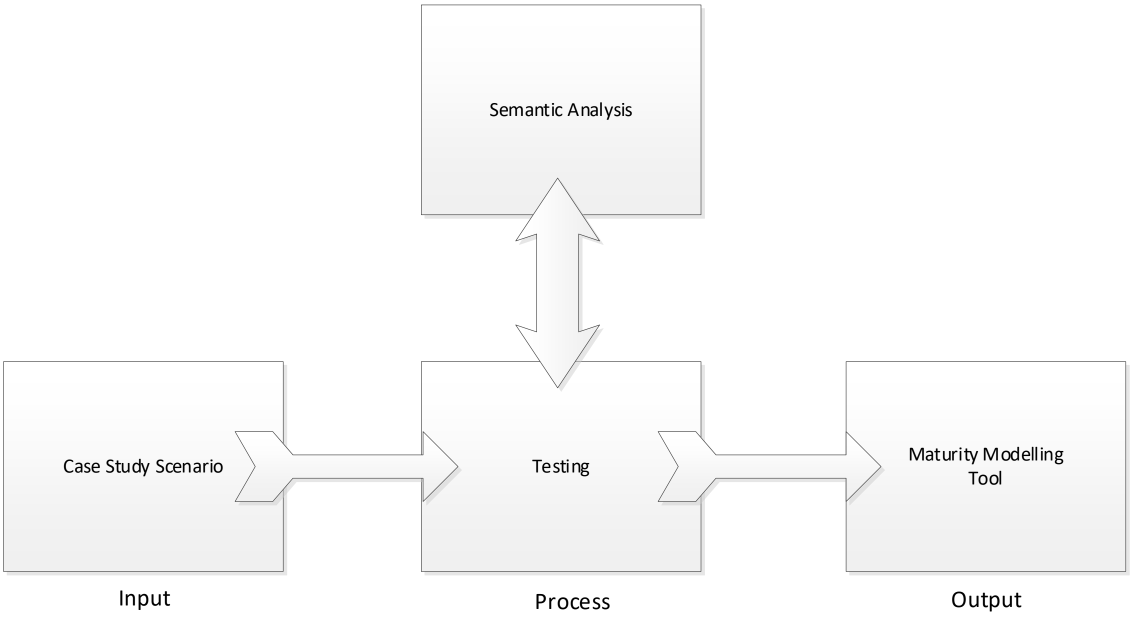


***Figure 3.9: Proposed Testing Process Model***

The objectives of this test phase as a design science method or practice as described in Section 3.2.2 will be discussed and evaluated for efficiency and utility. Process feedback is sought and integrated as an iterative process to determine the best practice, designed to manipulate and test the MM as a model and the taxonomy as a construct. Changes in the metrics that will also be evaluated for utility on the process feedback information will be reintegrated into the objective and design / development steps of this research phase. This iterative demonstration and feedback integration process is expected to occur at least twice. The resulting output from the validation testing process will be used to inform research phase six, the Maturity Model creation.

### 3.3.6 Phase Six: Maturity Model Creation

As determined in Chapter two, the identified research problem is the difficulty in assessing risk as a focus identification within the IoT domains. The solution postulated by this research is to provide a Design Science instantiation artefact in the form of a Maturity Model that is automatically generated.

### 3.3.7 Phase Seven: Thesis Writing.

The final phase of the research is to document the findings, discuss the results, identify limitations, and suggest areas for future research. This is the final DS process step of knowledge dissemination. The construction of the thesis documents has the complete process from inference, theory, application, analysis and then to knowledge. The literature review section is designed to provide problem identification and to assign a level of importance to the identified problem. The literature review component of the thesis identifies the current research within the identified domain of IoT risk enumeration. It also investigates foundational aspects of the identified domain and allows the researcher to create an inference from the analysis of other research outputs. The resultant inference is put forward as an artefact output definition and elaborates a novel research contribution in Chapter two. Chapter three defines and itemizes each step of the research process, which provides a list of process activities that encourage future research by providing a valid and repeatable research theory and practice method.

## 3.4 RESEARCH QUESTIONS

The study has been designed to test the following two hypotheses:

### 3.4.1 H1: Hypothesis One

Risk aspects are identified using cyber forensic and data analysis techniques.

### 3.4.2 H2: Hypothesis Two

The output from Hypothesis One testing informs a Risk Maturity Tool to identify IoT risk.

The results from the Hypothesis testing will provide answers to the following research sub-question:

### 3.4.3 RSQ1: Research Sub-Question One

What risk aspects are identified using cyber forensics and semantic data analysis techniques?

### 3.4.4 RSQ2: Research Sub-Question Two

Which risk inputs inform a Risk Maturity Tool for the Internet of Things?

The information gained from investigating the answer to RSQ1 will, in turn, provide answers to the Research Question:

### 3.4.5 RQ: Research Question

What factors improve Capability Maturity Risk Modelling for the Internet of Things?

## 3.5 DATA REQUIREMENTS

The data collection and evaluation components of this research are presented in Section 3.5. The objective of this section is to demonstrate the data collection methods and to indicate how the collected data will be used to answer the research questions given in Section 3.4. The integration of case study information has been selected for use in the data gathering element of this research.

Section 3.5 begins by providing information about rationale for the Proof-of-Concept use of case study analysis within this research. Section 3.5.2 evaluates the application of case study research; Section 3.5.3 identifies the risk of case study research when considering generalizations and provides mitigation strategies. Section 3.5.4 investigates the limitations of case study use within research.

### 3.5.1 The Use of Case-studies as Proof-of-Concept

An important aspect of the investigation and analysis of case studies within research is the reliance upon theoretical concepts. This enhances the research undertaken through the use of preliminary concepts, especially during the first stages (Yin, 2011). This approach will allow the case study analysis results to be placed within the appropriate literature review structure, as discussed in Section 2. The purpose of the Proof-of-Concept is to check the initial hypothesis and perception of the research direction identified. The analysis of the three preliminary case studies within this section will then provide information that will be used to further develop the knowledge and understanding of the topic of cyber forensic capability maturity enumeration. The development process will be enhanced through the identification of variables of significance and relevant data to be collected. Therefore, the following case studies, as an element of case study theory, are investigated as exploratory case studies, as identified by Yin (2011).

The use of exploratory case studies as a Proof-of-Concept, as applied within this research is justified as an illustrative example of the feasibility of the concepts postulated. More importantly, the use of the case studies as exploratory when defining theoretical considerations within this research provide a blueprint and the underpinning of a taxonomical structure that can be evaluated manually. The results from the manual evaluation is then used to help test the study questions and form the basis of the methodological needs. This approach also adds to the validity of the research, as the taxonomical groundwork provided through the exploratory case study investigations have not been predetermined and are identified through the case study analysis.

The disparate nature of the three exploratory case studies selected is intentional. The cases are unrelated and distinct by design and are many years apart in time, and also encompass different aspects of the IoT cyber forensic context. The first exploratory case study investigated involving Industrial Control Systems (ICS) occurred nearly twenty years before this exploratory analysis, in early 2000. The second exploratory case study, involving a Heating, Ventilation and Air Conditioning (HVAC) exploit of a Point of Sale (POS) system occurred six years before this exploratory analysis, in 2013. The third and final exploratory case study occurred within the last three years, in 2016. The information arising from each of the exploratory case studies will provide risk indicators and factors of interest that will be tabulated and presented as a taxonomical classification. The rationale of this approach is that the research will provide insights into relationships and designations of risk that will apply to each of the case studies individually and to all of the case studies collectively.

The use of case studies to evaluate and assess the effectiveness of the proposed methodological approach in creating the artefact outcome has been shown to be valid as a research approach Creswell (2014) who found case-studies can be used to provide insight into activities and processes (p. 43). Yin (2011) observes that in case-studies questions are directed at the researcher rather than the artefact output, and therefore, as the research progresses research questions are updated with the insights obtained from incremental learnings. This is appropriate as it conforms to the design science concept of continual iterative artefact improvement. Yin (2011) indicates that the input process can derive from a line of enquiry (e.g., methodological framework) but does not necessarily come from a verbatim script. On that basis Yin's (2016) case-study method is better suited to Proof-of-Concept of the method and instantiation as the artefact can be updated as incremental learnings progress.

### 3.5.2 Case-studies

Later, Yin (2014) observes there are two critical steps in developing a case-study methodology; defining the case; and bounding the case. Yin (2014) identifies five potential rationales in case-study design with two rationales directly applicable to this research for data gathering purposes. The first rationale is *critical* (p. 52) which notes that as experimentation progresses the results make alternative sets of explanations that may apply in the exploratory stage. The second rationale is *revelatory* (p. 52) for which Yin (2014) cites examples where the findings were different from the initial expected outcome. Therefore, applying Yin's (2014) two rationales to this research the methodological process may be designed to support an initial procedure at its outset, but with the flexibility to update the process as the research progresses. Thus, the proposed methodological process or instantiation should ensure a consistency of measurement across the case-studies.

### 3.5.3 Generalizations

A risk identified by Yin (2014) in adopting a case-study methodology is that the findings of a single case-study can be generalized and may be applied too broadly. In this research, observing the three case studies selected across the wide and diverging applications have been designed to reduce the risk of generalized findings. Eisenhardt (1989) considers the optimal number of case-studies necessary to avoid generalisation is between 4 cases and 10 cases. However, this Proof-of-Concept is a small-scale research pilot study where a case-study observation involves the output and creation of viable taxonomy for the purposes of creating a capability maturity model. For observations where the finding could be considered subjective, the three case-studies will be consistent with the views of Eisenhardt (1989). To ensure generalizations have validity, this research needs to be rigorous enough to identify factoids and hype used by marketing departments amongst others to drive sales rather than the more usual observed levels of engineering risk and uncertainty.

### 3.5.4 Limitations

While a research design will need to be adaptive so that it can be updated with incremental results, Yin (2014) cautions that adaptive processes should not lessen the rigor which case-study procedures follow. Multiple investigations are identified as providing two key advantages; firstly, enhancing the creative potential of the investigator team with complementary insights, and secondly, convergence of insights from multiple investigations boosts confidence levels (Eisenhardt, 1989). As this is a small-scale pilot study, designed to provide a Proof-of-Concept, the findings are those of a single researcher.

## 3.6 DATA ANALYSIS

Analysis of data generated throughout this research process is a multilayered process. Each output component is in the form of an artefact as an integral element of the design science research process. Each artefact output is analyzed, and the findings are presented to expert evaluation for potential artefact improvement. The analysis process involves comparing the output results with aspects of test case studies, as discussed in Section 3.5.

Section 3.6.1 describes how case study structure will be used in this research as a distinct stand-alone analytic. It will provide single points of information that will inform the testing process. Section 3.6.2 describes how the data will be analyzed, and the results demonstrated and analyzed for design improvement and evaluation as part of the design science research process.

### 3.6.1 Case-study Structure

The need for discipline in the observation process is stressed by Baxter and Eyles (1997). They consider that while qualitative researchers are encouraged to allow the situation to guide the research procedures. For the research to be evaluated there must be a clarity of design and transparency in the derivation of findings. Baxter and Eyles (1997) identify research practices that assist transparency including the use of standardized guides. Eisenhardt and Graebner (2007) note each case serves as a distinct experiment that stands on its own as an analytic unit. In the context of this research, an analytic unit refers to the components that assist the formation of the artefact instantiation and the Proof-of-Concept. Adopting the insights of Baxter and Eyles (1997) for research discipline, a standardized observational structure setting out the risk model taxonomy will be used to record each case-study. Applying the lessons of Daley (2004), themes identified during research will then be tied to the instantiation in the form of a taxonomical structure and produced as a concept map to compare similarities or differences.

In summary, Eisenhardt and Graebner (2007) assists an understanding that each case-study is a distinct experiment that stands on its own as an analytic unit. However, Yin (2014) cautions that the findings of a single case-study must not be generalized. Hence, the number of case-studies completed should be sufficiently layered to ensure research validity. This critical evaluation of research methodology has determined theory building from qualitative data is both legitimate (Eisenhardt & Graebner, 2007, p. 25), and is ideally suited to this pilot study.

### 3.6.2 Analysis of Findings

The findings data will be analyzed, and the outcome of the analysis will be evaluated for utility and efficiency, and the output will be integrated into a design science feedback loop. The analysis process will integrate information derived from the data from the automatic Semantic Analysis Engine (SAE) taxonomy output, and then compare the data output with a manual analysis of case study information. Changes of the SAE input, in the form of single dependent variable manipulation will then be analyzed for change in the taxonomic output. The analysis of the effects of the change upon the taxonomic output will then be evaluated for utility and efficiency to identify potential design improvement. The evaluation information will be incorporated into the design and development stage. The integration of design improvement information forms the underlying strength of the design science research methodology utilized within this research.

## 3.7 CONCLUSION

Chapter three began by describing the theory of taxonomy creation methods, and the importance of taxonomic classification processes underpinning this research. The primary section then developed a practical taxonomic creation process, describing the process steps undertaken. The steps include a text input selection process, the semantic analysis engine process, and the taxonomy creation process. Each step is described in detail, demonstrating each integral component of the taxonomy creation process that is used in this research. Section 3.2 presents DS as the design methodology selected for use within this research and discusses the relevance to the proposed research and the reliability that iterative efficiency evaluation input adds. The research process sequence undertaken is then described as activities and outlined against the design science process sequence. A sequence of artefacts is defined and compared to a design science abstraction. Each artefact is then described in detail, and each output type is discussed. The research phases undertaken are described, outlining the objectives, design and development components of the design science process. Two hypotheses are then postulated, which are designed to answer the research sub-question. The resulting information provides evidence to answer the research question: What factors improve Cyber Forensic Readiness Capability Maturity for the Internet of Things?

Data requirement components are then discussed, showing the importance of case study evaluation. It defines the risk of generalizations and the limitations of using case study data analysis.

Chapter four presents a pilot study and Proof-of-Concept of the processes outlined in Chapter three. Chapter four is designed to provide a manual feasibility demonstration, which investigates the practical potential of the underlying theory and models. It manually applies the process steps (automation is reported in Chapter six). The output demonstrated in Chapter four represents the first research phase defined in Section 3.3.1 and presents a DS research output artefact in the form of an inference. The inference artefact is a contribution resulting from this research and demonstrates the transition sequence from a generalization to solve an identified problem, to an actual implementation of the proposed solution.

# Chapter 4
# PILOT STUDY

## 4.0 INTRODUCTION

Chapter four demonstrates a manual process to populate a risk attribute taxonomy. Three case study scenarios from different IoT ecosystems are tested using a manual taxonomy creation process as a pilot study and demonstration of the prototype proposed. The first case study is based on a security attack in an Industrial Control System (ICS) infrastructure. The second case study investigates the exploitation of a distributed commercial network system, and the third case study is an Autonomous Vehicle accident. Thus, the test case scenarios represent divergent cases involving real examples of IoT vulnerability and risk. Each test case is selected due to the completeness of information available regarding the identification of the risk aspects. Each case has been involved in a litigation process or subject to an official investigation. The litigation process indicates that the information gathered is legally admissible evidence, and therefore the information is based on testable assertions. Each IoT case study attack scenario provides the input for the Proof-of-Concept, and the feasibility testing of the proposed taxonomical creation steps, as shown in Figure 4.1. The attack / test case scenarios will each be analyzed for risk attribution. The results of the analysis will then be processed through a manual application of the proposed taxonomy creation procedure. The taxonomy output will then be tested for Proof-of-Concept.

The Proof-of-Concept demonstrated in this chapter is delivered through the application of the process steps shown in Figure 4.1. The process steps form part of the Taxonomy Creation Tool method as discussed in Section 3.1. The first step requires an input of information, selected from professional institutions, as discussed in Section 2.5, and shown in Figure 4.1, termed ‘Information Links.’ The information input, for the purposes of the Proof-of-Concept, is selected from the National Institute of Standards and Technology (NIST), which provides the information vocabulary. The vocabulary is then used to analyze the test case scenarios, forming the taxonomy design and population.

Chapter four is structured as follows: Section 4.1 presents the establishment of information links and then demonstrates the domain and family categorization process. Section 4.2 provides two examples of the manual application of the taxonomy creation tool. The following three Sections; 4.3, 4.4 and 4.5 each apply the taxonomy creation tool manually to the three case studies selected for analysis. Finally, Section 4.6 discusses conclusions drawn from the manual taxonomy tool application as demonstrated in this chapter.

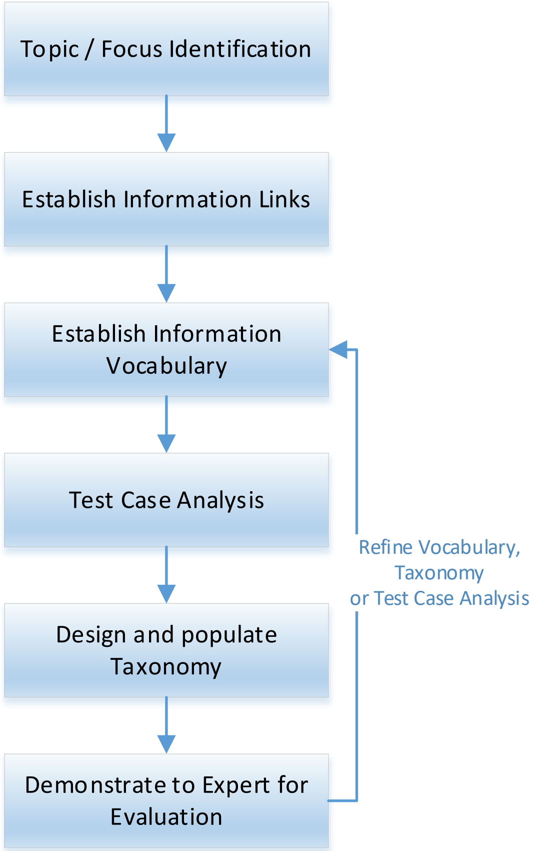

Topic / Focus Identification
Establish Information Links
Establish Information Vocabulary
Test Case Analysis
Design and populate Taxonomy
Demonstrate to Expert for Evaluation
Refine Vocabulary, Taxonomy or Test Case Analysis

*Figure 4.1: Taxonomical Creation Process Steps*

## 4.1 TAXONOMY STRUCTURE

As discussed in Section 2.5 and Section 3.2, the taxonomical entity classification organization is based on the following process, as outlined in Figure 4.2. The Top-Level hierarchical structure is interest based and is determined as the 'identifier' value. The labels are selected to form a parent-child relationship that will aid the identification of the cause/effect associations. As shown in Figure 4.1, the establishment of the information link is an essential component to provide best practice and standards terminology, for the Taxonomical process utilized in this research. Once the identifier and information links have been established, the vocabulary can be controlled and the labels for the second level, 'Domain' and third level 'Family' entities are selected and applied. Finally, upon analyzing the test cases, at the lower level 'Attribute' entities are assigned, and the taxonomical structure is populated.

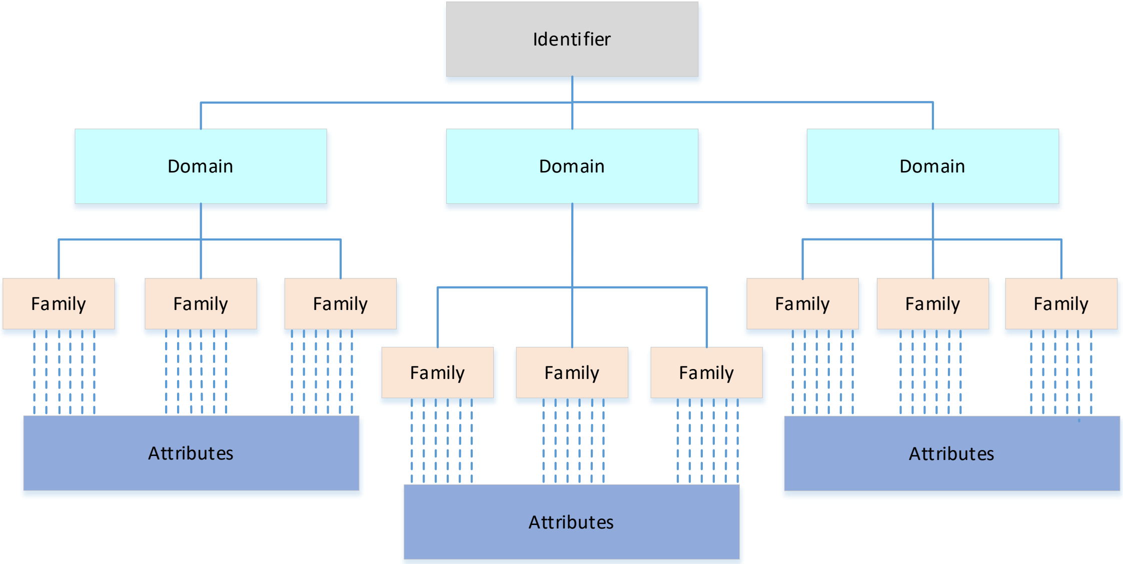


***Figure 4.2: Taxonomy Entity Structure***

Thus, the process begins with the selection of an identifier, which defines the overarching topic of the Taxonomy. This then determines the focus of the literature to be investigated, in order to form the Domain and Family entity terms. This process is discussed in Sub-Section 4.1.1 and 4.1.2. The Attribute entity terms are allocated upon the analysis of the three case studies. Therefore, following the process established in Chapter three and shown in Figure 4.2, the researcher is able to control and select the vocabulary and thus the Domain and Family entity terms. The vocabulary selection process begins with the allocation of a field or discipline area, determined by the 'Identifier' which then provides information for the information input determination.

### 4.1.1 Identifier and Information Links

The identification and assignation of a topic or focus, determines the subject area from which a contribution is sought, as discussed in Section 2.5 and Section 3.2. The subject area information input allows the researcher to establish the information vocabulary and thus control the selection of entity terms.

Security, and the associated security requirements has been chosen as the specific area of identification for this case study analysis. This process meets the guidelines and requirements as an illustrative example to provide the creation process of the proposed taxonomical structure and will also provide a robust manual evaluation process as discussed in Section 2.5.1.2 and Section 3.2.2. The Information Links are selected from NIST which provides information in the form of standards documentation. This is utilized to provide terminology vocabulary for the test case Proof-of-Concept creation for a risk identification taxonomy.

The following documents have been selected from the NIST repository: NIST FIPS 200, which provides information regarding the minimum security requirements for the protection of federal information and information systems, (NIST 200, 2006), and NIST special publication SP800-53, Security and Privacy Controls for Information Systems and Organizations, providing a control selection process designed to protect organizational assets and manage organizational security risk (NIST 800-53, 2020).

### 4.1.2 Information Vocabulary

The information vocabulary utilized to populate the lower-level entity taxonomical classification labels is determined from the information link input as discussed in Section 4.1.1. The vocabulary is carefully selected from the information input of two NIST professional guidelines. The controlled information vocabulary pool thus collected from the information link input is then used to further populate the taxonomical structure. The lower-level taxonomical classification labels have risk entities, The Domain, Family and Attribute (see Figure 4.2). These are determined from analysis of the three case studies and using the Information Vocabulary pool. The process of assigning labels to the Domain and Family entities is demonstrated.

The first entity of the hierarchical structure that is populated, is the third level, or Family entity. This process was defined in Chapter two, where the third level of the Taxonomical structure is used to determine the second level. The Family entity categorization selected from the information vocabulary is shown in Table 4.1.

***Table 4.1: Family Categorization***

| Family |
|---|
| Certification |
| Awareness and Training |
| Audit and Accountability |
| Certification, Accreditation, and Security Assessments |
| Configuration Management |
| Contingency Planning |
| Identification and Authentication |
| Incident Response |
| Maintenance |
| Media Protection |
| Physical and Environmental Protection |
| Planning |
| Personnel Security |
| Risk Assessment |
| System and Services Acquisition |
| System and Communications Protection |
| System and Information Integrity |

Further evaluation of the selected Family entity terms and utilizing the information vocabulary established from the NIST information link input, determines a categorization structure. The categorization process, as discussed in Section 2.6, is then used to carefully select the Domain entity terms. The following Domain or second level entities of the Proof-of-Concept taxonomical structure have been ascertained to be: Management, Operational, and Technical, as shown in Table 4.2. The Domain entity classifications also align with the previously determined top level or Identifier notification of Security.

***Table 4.2: Domain Categorization Additions***

| Domain | Family |
|---|---|
| Management | Certification, Accreditation, and Security Assessments |
| Management | Planning |
| Management | Risk Assessment |
| Management | System and Services Acquisition |
| Operational | Awareness and Training |
| Operational | Configuration Management |
| Operational | Contingency Planning |
| Operational | Incident Response |
| Operational | Maintenance |
| Operational | Media Protection |
| Operational | Physical and Environmental Protection |
| Operational | Personnel Security |
| Operational | System and Information Integrity |
| Technical | Access Control |
| Technical | Audit and Accountability |
| Technical | Identification and Authentication |
| Technical | System and Communications Protection |

The three Domain entity classification are then separated, and the vocabulary is processed and analyzed a final time, to determine singular entity values. Hence, the selection is controlled which is an essential concept of taxonomical structure creation. This concept, as part of the taxonomical Creation Steps is discussed in depth in Section 2.5. The separation of the three domain level entities is shown in the following three Tables. Table 4.3 demonstrates the final Family entity labels for the Technical Domain, Table 4.4 demonstrates the final Family entity labels for the Operational Domain, and Table 4.5 demonstrates the final Family entity labels for the Management Domain.

*Table 4.3: Technical Domain Categorization*

| Domain | Family |
|---|---|
| Technical | Access Control |
| Technical | Identification |
| Technical | Authentication |
| Technical | Event Logging |
| Technical | System Protection - Physical |
| Technical | System Protection - CyberSecurity |
| Technical | Communications Protection |

*Table 4.4: Operational Domain Categorization*

| Domain | Family |
|---|---|
| Operational | Audit and Accountability |
| Operational | Awareness |
| Operational | Training |
| Operational | Configuration Management |
| Operational | Contingency Planning |
| Operational | Incident Response |
| Operational | Maintenance |
| Operational | Media Protection |
| Operational | Physical Protection |
| Operational | Environmental Protection |
| Operational | Personnel Security |
| Operational | System Integrity |
| Operational | Information Integrity |
| Operational | Confidentiality |

*Table 4.5: Management Domain Categorization*

| Domain | Family |
|---|---|
| Management | Certification |
| Management | Accreditation |
| Management | Security Assessments |
| Management | Planning |
| Management | Risk Assessment |
| Management | System Acquisition |
| Management | Services Acquisition |

## 4.2 MANUAL TAXONOMY CREATION PROCESS EXAMPLES

Section 4.2.1 and Section 4.2.2 provide an example of the analysis process that is followed in each of the three case studies. An analysis of aspects of security risk will be provided through a descriptive paragraph. The information is then parsed through the Information Vocabulary and compared with the Family entities determined in Table 4.3, Table 4.4 and Table 4.5. This populates the Risk Attribute entities within the Proof-of-Concept taxonomical structure.

### 4.2.1 Demonstration of Taxonomy Application: Stuxnet

The attack vectors to which SCADA systems have been subjected, as part of a continual growth of attacks on industrial control systems, have been increasing in volume and complexity since the Stuxnet case occurred. Examples of the complexity of historic attacks include Stuxnet, which was uncovered in 2010. The Stuxnet attack not only caused severe damage to critical components within a nuclear enrichment plant in Iran but was also associated with infecting an estimated 200,000 computers worldwide. The Stuxnet worm infected many PLC controllers, mainly produced by Siemens, throughout Europe and in particular, Germany. The increasing Internet connectivity implementations within the Industrial Control System area has become a cause of concern, especially when considering the implications for what is now termed ‘Critical Infrastructure’. Table 4.6 shows the taxonomical process output example using Stuxnet data and applying the creation and resolution processes defined in Chapter two.

***Table 4.6: Taxonomy Creation Process Example - Stuxnet***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection | Infrastructure |
| Technical | Communications Protection | Infrastructure |
| Operational | Configuration Management | Planning |
| Operational | Contingency Planning | Complexity |
| Management | Risk Assessment | Implications |

### 4.2.2 Demonstration of Taxonomy Application: Executive Order 13636

The concern for vulnerabilities in critical infrastructure in the United States, prompted the then President Barack Obama to draft an Executive Order, 13636, on February 12, 2013 which specifically addressed the need for comprehensive security implementations to be considered within US critical infrastructure (NIST 2014). An important aspect of Executive Order 13636, when viewed in the context of this research, is that the term ‘Critical Infrastructure’ is clearly defined as “systems and assets, whether physical or virtual, so vital to the United States that the incapacity or destruction of such systems would have a debilitating impact on security, national economic security, national public health or safety, or any combination of those matters”. The importance of this definition for research is that it demonstrates the requirement for a clear definition to be accepted by all parties, when referring to the term, thus the importance of the development for a comprehensive taxonomical structure for the IoT. Table 4.7 shows the taxonomical process output example using Executive Order 13636 in the case study analysis.

*Table 4.7: Taxonomy Creation Process Example - Executive Order 13636*

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection - Physical | Assets / Infrastructure |
| Technical | System Protection – Cyber Security | Virtual |
| Operational | Configuration Management | Description / Definitions |

## 4.3 TEST CASE 1: MAROOCHY SHIRE SEWAGE SPILL

The Maroochy Shire Sewage spill which occurred in March 2000, has been selected as the first test case scenario. The case has been cited world-wide since the date of the attack and the perpetrator has been prosecuted, thus the information has been subjected to evidence admissibility challenges. The quality of evidential information is a key point when selecting incidents deemed suitable for case studies to be included within this research. Also, the incident has been analyzed and reported in many forums as an example of the risks inherent in a Supervisory Control and Data Acquisition (SCADA) Industrial Control System (ICS) that received insufficient preparation for security considerations during the installation process.

### 4.3.1 Analysis

With a population of 130,000 in a 1,160 km$^2$ area, Maroochy Shire is a Queensland, Australian small town. The Maroochy Shire Council operated a Sewage Treatment Plant using a Supervisory Control and Data Acquisition (SCADA) system to control 142 Pumping Stations processing 35 million liters of effluent per day. A new automated SCADA based industrial control system was tendered, and a local company won the contract to install the system, designed to centralize control of the entire treatment system.

### 4.3.2 SCADA System Installation

Maroochy Shire Council appointed Hunter Watertech (HWT) as contractors, to upgrade the Sewage Treatment Plant. HWT installed a Remote Terminal Unit / Programmable Logic Controller (RTU/PLC) system, called a HWT PDS Compact 500 RTU, into each of the 142 Pumping Stations in 1997. The automated control system implemented by HWT within each pumping station communicated with a centralized control center. The control center altered flow through signals that changed the parameters for regulation of the individual pumps inside each station. A SCADA system and the Cyber-physical systems at the pumping station has actuators receiving commands. This then transmits information from the sensors within the various valves, pumps and pipes, back to the control center, conveying system feedback information through an access point contained within each pumping station. The communication transport layer was a primitive radio frequency system provided by a private two-way radio system of radio operators and repeating stations. Thus, the control of each pumping station was not limited to transmissions from the control center, but in actuality was controlled through one of the pumping station's access points. Table 4.8 shows the installation risk attribute identification population for the risk taxonomy attribute entities.

***Table 4.8: Installation Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection - Physical | Infrastructure |
| Technical | Communications Protection | Infrastructure |
| Operational | Awareness | Security |
| Operational | System Integrity | Security |
| Management | Planning | Audit |
| Management | Risk Assessment | Audit |
| Management | Security Assessments | Audit |

### 4.3.3 Fault Condition

After the installation process was completed in mid-January 2000, the SCADA system was tested and brought online, and it functioned correctly. However, in late January 2000, the SCADA system started experiencing fault conditions. The faults included configuration changes, loss of pump control, false alarms, and communication outages. Initially, it was thought that the problems were being caused by the SCADA system itself, so the system was re-installed and re-tested. This, however, did not solve the issues, and the faults continued unabated. These problems continued throughout February until mid-March. As the SCADA system installed by HWT did not process any logging information and was only setup to exchange simple messages such as 'pump running' and 'tank full'. Hence, the system engineers were finding it difficult to trace the underlying cause of these faults (Mustard, 2006). Table 4.9 shows the fault condition risk attribute identification population for the risk taxonomy attribute entities.

***Table 4.9: Fault Condition Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Event Logging | Incident identification |
| Operational | Configuration Management | Control |
| Operational | Incident Response | Mitigation |
| Operational | Awareness | Control |
| Operational | Contingency Planning | Continuance |
| Management | Risk Assessment | Prioritization |

### 4.3.4 Forensic Investigation

An employee of HWT was appointed to investigate the problems specifically, and the employee installed a logging program into the SCADA system, which began to provide useable forensic information on March 16, 2000. The logging program was designed to capture and log information such as radio traffic along with control information not only from the central command center, but information received by each of the pumping station local access points. This was the first implementation of a process designed to perform specific Cyber Forensic data capture and logging. Within a month, the investigators determined that the logged evidence indicated the faults were being generated through human intervention rather than failure at a hardware level (Mustard, 2005). Table 4.10 shows the forensic investigation risk attribute identification population of the risk taxonomy attribute entities.

***Table 4.10: Forensic Investigation Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Event Logging | Analysis |
| Technical | Communications Protection | Control |
| Technical | System Protection – Cyber Security | Intrusion Protection |
| Operational | Incident Response | Procedure |
| Operational | Configuration Management | Documentation |
| Operational | System Integrity | Control |
| Operational | Awareness | Analysis |

### 4.3.5 Forensic Evidence

Evidence presented at the District Court of Maroochydore as described in the Queensland Supreme Court of Appeal transcript (Supreme Court of Queensland R v Boden Vitek, 2002) ascertained the following specific information regarding the Cyber Forensic information gathered by the investigators and the determinations upon the analysis of the information by the forensic investigators. Unauthorized access was gained, that was designed to alter electronic data of computers that controlled the Maroochy Shire Council’s sewage pumping stations, causing operational malfunctions to occur. The primary investigator ascertained that a specific pumping station was the source of the corrupted messages that were causing the faults. The investigator checked the pumping station’s system physically and found that the system was functioning correctly and therefore eliminated the possibility of physical tampering or hardware systems failure causing the issues. All the corrupted messages were shown to be associated with ID 14, which was the unique identification number of the SCADA controller associated with that pumping station. Acting upon this knowledge, the investigator changed the identification number of the SCADA system of that particular pumping station to the number 3. This would enable the investigator to identify legitimate information, because the messages would identify themselves as now coming from pumping station 3. More importantly, this renaming of the pumping station identification would determine that any information associated with an identification number 14, would be easily shown to be falsified. Table 4.11 shows the forensic evidence risk attribute identification population of the risk taxonomy attribute entities.

***Table 4.11: Forensic Evidence Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Access Control | Accountability |
| Technical | Identification | System Access |
| Technical | Authentication | System Access |
| Technical | System Protection - Physical | Infrastructure |
| Operational | System Integrity | Control |
| Operational | Information Integrity | Control |
| Operational | Configuration Management | Documentation |

### 4.3.6 Attack Penetration

This approach was successful for a short time, but the intruder gained control of the system and altered the malicious data identification number to 1, which was a legitimate identification number elsewhere on the SCADA system, causing more serious issues. The malicious intruder altered data so that the central computer was unable to exercise correct control, and technicians had to be mobilized across the entire system to correct faults at the affected pumping stations. On April 19, analysis of the forensic information determined that the malicious program causing the issues had been run at least 31 times since February 29. The forensic information captured was ultimately able to ascertain that on 23 April the assailant began attacking the SCADA system at 19:30 and concluded at 21:00, disabling alarms of four pumping stations through the use of data, using the identification number of pumping station number 4. Table 4.12 shows the attack penetration risk attribute identification population of the risk taxonomy attribute entities.

***Table 4.12: Attack Penetration Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection | System Access |
| Technical | Access Control | Accountability |
| Technical | Identification | System Access |
| Technical | Authentication | System Access |
| Operational | Configuration Management | Documentation |
| Operational | System Integrity | Control |
| Operational | Information Integrity | Control |

#### *4.3.6.1 Catastrophic Environmental Failure*

The most grievous fault generated by this continuation of malicious activity, caused a catastrophic failure that caused 800,000 liters of raw effluent to be released, which affected residential housing, parks and the local river. Table 4.13 shows the catastrophic environmental risk attribute identification population of the risk taxonomy attribute entities.

***Table 4.13: Catastrophic Environmental Failure Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection - Physical | Catastrophic Damage |
| Operational | Environmental Protection | Adverse effect on living organisms |
| Operational | Configuration Management | Control |
| Operational | Incident Response | Mitigation |
| Management | Risk Assessment | Contingency |

### 4.3.7 Post Incident Malicious Employee Forensic Evidence

The instigator of the malicious attacks had, by the 23rd April 2000, already fallen under suspicion. A vehicle that was owned by the suspect, was stopped by police, and searched. Inside the vehicle an HWT PDS Compact 500 SCADA computer, a two-way radio, a laptop and system specific cabling were located. Upon examination of the computer equipment by a police computer expert, the following information was presented as evidence, which was unchallenged by the accused. The software installed in the laptop was software developed by HWT, designed to alter the configuration of the HWT PDS Compact 500 SCADA computers. This software had no other practical use, other than to access the Council's sewerage system, as installed by HWT. The logs contained within the Laptop showed that the software had been used at least 31 times between 7 April and 19 April. The startup and shutdown times of the software were consistent with the times of the attacks logged by the forensic capture system of HWT. The laptop logs also showed that the software was last used at 21:31 on 23 April 2000. The two-way radio located in the accused's vehicle was set to the specific frequencies of the repeater stations used by the SCADA system's private radio communication system. The cabling located in the accused's vehicle were necessary to connect the PDS Compact 500 SCADA computer to the laptop and the two-way radio. The PDS Compact 500 SCADA computer had been altered to identify itself as pump station 4. Table 4.14 shows the malicious employee risk attribute identification population for the risk taxonomy attribute entities.

***Table 4.14: Malicious Employee Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection - Physical | Audit |
| Technical | Communications Protection | Monitor |
| Technical | Event Logging | Monitor |
| Technical | Access Control | Revoke |
| Technical | Identification | Revoke |
| Technical | Authentication | Revoke |
| Operational | Configuration Management | Documentation |
| Operational | Audit and Accountability | Hardware / Equipment |
| Operational | Personnel Security | Employment Termination |
| Operational | System Integrity | Monitor |
| Operational | Awareness | Employment Termination |
| Management | Security Assessments | Employee |
| Management | Risk Assessment | Employee |

### *4.3.7.1 Malicious Employee Court determination*

The Court determined, however, that the evidence presented did not directly prove that the accused had sent the maliciously altered messages which corrupted the Council's sewerage operation. Nevertheless, as the evidence presented was unchallenged by the accused, and the accused did not present any evidence in opposition, the Court found that the evidence formed the foundations of a "*strongly circumstantial case*" (Supreme Court of Queensland R v Boden Vitek, 2002). The absence of evidence or explanation by the accused, the conclusions drawn by the jury of the "*almost overwhelming*" case presented by the Crown, enabled the jury to readily return a guilty verdict. The sentencing judge found that the accused was aware that the overflows and environmental harm could occur, even though it was not the accused's "*sole purpose to cause sewerage overflows*" (Supreme Court of Queensland R v Boden Vitek, 2002).

***Table 4.15: Malicious Employee Court Determination Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection - Physical | Malicious intent |
| Technical | Event Logging | Analysis |
| Technical | Access Control | Breach |
| Technical | Identification | Breach |
| Technical | Communications Protection | Breach |
| Operational | Configuration Management | Documentation |
| Operational | Contingency Planning | Employee Termination |
| Operational | Incident Response | Prosecution |
| Management | Risk Assessment | Audit |
| Management | Security Assessments | Audit |

The accused was convicted of willfully and unlawfully causing serious environmental harm and also of 27 counts of using a restricted computer to cause detriment or damage.

The accused was later sentenced to two years in custody (Abrams & Weiss, 2008). Table 4.15 shows the malicious employee court determination risk attribute identification population of the risk taxonomy attribute entities.

#### *4.3.7.2 Malicious Employee Scrutiny*

The attacks experienced by the Maroochy Shire Sewage Treatment system, and the underlying SCADA industrial control system was carried out by a maliciously motivated, and highly skilled and knowledgeable ex-employee, who had been a trusted contracted employee. The employee was involved in the HWT SCADA system installation project at the Maroochy Sewerage Treatment plants from late 1997 until December 3rd 1999. He was employed under contract as a site supervisor for the project. He was involved with every aspect of the installation and at each of the 142 pumping stations. In December 1999, the employee approached the Maroochy Shire Council for employment, and (it appears) that he expected this application to be viewed favorably, but he was told by the Maroochy Council to re-apply at a later date. The employee then had a disagreement with HWT and resigned, citing a "*strained relationship*" (Abrams & Weiss, 2008). However, when the (now) ex-employee reapplied to the Maroochy Council, he was informed that the Council would not be requiring his services.

***Table 4.16: Malicious Employee Scrutiny Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection - Physical | Equipment |
| Operational | Audit and Accountability | Equipment<br>Access<br>Control |
| Operational | Awareness | Termination Procedure |
| Operational | Training | Termination Procedure |
| Operational | Configuration Management | Revoke Access |
| Operational | Contingency Planning | Employee Role |
| Operational | Incident Response | Prosecution |
| Operational | Maintenance | Revoke Access |
| Management | Security Assessments | Assess Access |
| Management | Planning | Termination Procedure |
| Management | Risk Assessment | Employee Role |

This appears to be the underlying motive, where the ex-employee of HWT decided to get even with both HWT and the Maroochy Council (Slay & Miller, 2007). Table 4.16 shows the catastrophic malicious employee scrutiny identification population for the risk taxonomy attribute entities.

## 4.4 TEST CASE 2: TARGET PRIVATE INFORMATION DATA BREACH

Through the acquisition of security credentials from a Heating Ventilation and Air Conditioning (HVAC) installation, hackers were able to gain access to the United States retailer Target Corporation's (Target) internal network (Manworren, Letwat, & Daily, 2016). The hackers subverted Target's Point of Sale (POS) system and captured 40 million credit card numbers along with 70 million pieces of personal information (Riley, Elgin, Lawrence, & Matlack, 2014). The Target data breach has been selected because of the global interest the attack generated, and the number of analyses available. The legal testimony is available and contributes to factual evidence presented for analysis. The Target test case also provides information where a risk analysis and security audit had been undertaken and mitigation controls applied prior to the breach.

### 4.4.1 Analysis

The technical difficulty of securing information and risk attribution analysis of large scale, distributed networks is challenging. The Target distributed network had installed software defense mechanisms, and instituted best practice recommendations. The network applied specific measures to guard the network from malware attacks and data exfiltration. The defense system functioned correctly but several timely warnings were ignored, and some system functionality was deactivated by system administrators.

### 4.4.2 Security Defense Installation

Target implemented a range of data protection safeguards, as well as employing a dedicated internal staff team of security professionals. Target was audited in September 2013 for the Payment Card Industry Data Security Standard (PCI-DSS) and successfully passed this security audit. FireEye, a specialist cybersecurity company working in conjunction with the CIA developed and implemented a malware detection suite. Best practices for protecting payment card information on computer systems were followed and verified through audits carried out by both FireEye and the PCI-DSS. The security center was distributed across the globe and was designed to provide 24/7 cybersecurity threat monitoring. Table 4.17 shows the security defense installation risk attribute identification population for the risk taxonomy attribute entities.

*Table 4.17: Security Defence Installation Risk Attribute*

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Access Control | Compliance |
| Technical | Identification | Compliance |
| Technical | Authentication | Compliance |
| Technical | Event Logging | Best Practice |
| Technical | System Protection – Cyber Security | Malware Detection |
| Technical | System Protection - Physical | Security Team |
| Technical | Communications Protection | PCI DSS<br>Distributed Security Center |
| Operational | Audit and Accountability | Compliance |
| Operational | Information Integrity | Malware Detection<br>PCI DSS |
| Management | Certification | Compliance |
| Management | Accreditation | PCI DSS |
| Management | Security Assessments | FireEye |
| Management | Planning | Audit |
| Management | Risk Assessment | Internal Audit |

### 4.4.3 Third Party Service Supplier

A heating, ventilation and air-conditioning (HVAC) supplier was engaged by Target as a third-party service provider. They were a supplier of refrigeration devices and services and were working with Target to support an expansion of fresh food offerings. The HVAC Company relied upon a free non-commercial version of anti-malware software to provide their internal security. This anti-malware software did not provide real time protection. The breach of the HVAC supplier's system involved in an infection of malware, that was used to steal logon credentials from the HVAC Company's computer system. The credentials stolen then allowed the attackers remote access to the Target network. The credentials were designed for access to contract submission, project management and electronic billing processes. This appears to be the initial entry point of the breach to Target's network system. Table 4.18 shows the third-party supplier risk attribute identification population for the risk taxonomy attribute entities.

*Table 4.18: Third Part Supplier Risk Attribute*

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Access Control | External Supplier |
| Technical | Identification | External Supplier |
| Technical | Authentication | External Supplier |
| Technical | Event Logging | Analysis |
| Technical | System Protection – Cyber Security | External Security Processes |
| Technical | Communications Protection | External Access |
| Operational | Audit and Accountability | External Security Processes |
| Operational | System Integrity | Real time Protection |
| Management | System Acquisition | Notification |
| Management | Security Assessments | 3rd Party Audit Process |

### 4.4.4 Attack Penetration

Upon gaining access to Target's network system, the attackers discovered that there was no segmentation within Target system and therefore this allowed complete access across all the points of the network system. Thus, when access to Target's system was granted, the attackers were then able to escalate account privileges and traverse the network at will. Hence, the attackers were able to access mission-critical back-end systems, point-of-sale terminals and other devices. An external audit team post of an incident response was able to access a cash register after compromising a counter sale in the delicatessen department. It was also discovered by the audit team post incident that there was a problem with password policy enforcement. Many login credentials were able to be broken using standard password lists and rainbow tables. The use of weak passwords enabled the audit team to crack over 500,000 passwords, representing 86% of accounts of internal target systems. Table 4.19 shows the attack penetration risk attribute identification population for the risk taxonomy attribute entities.

***Table 4.19: Attack Penetration Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Access Control | Segmentation |
| Technical | Authentication | Segmentation |
| Technical | System Protection – Cyber Security | Border control |
| Operational | Audit and Accountability | Account Authorization Escalation |
| Operational | System Integrity | Account Authorization Escalation |
| Management | Security Assessments | Account Authorization Escalation<br>Border control<br>Segmentation |
| Management | System Acquisition | Border Control<br>Segmentation |

#### *4.4.4.1 Authorization Escalation*

Once system-wide access was granted, the authorization escalation allowed the attackers to install malware directly upon Point of Sale (POS) terminals. These were accessed, at a hardware, Random Access Memory (RAM) level. The RAM was a hardware component of the POS system, situated between the card swipe reader hardware and the encryption software application layer and contained unencrypted card information. The attackers were able to scrape data from the RAM and capture the unencrypted information of cardholder data before the data was encrypted at another section of POS terminal. Thus, the attackers were able to access this unencrypted data, encapsulate the data, and then send the data out of the POS system, for further transmission. The RAM component of the POS system, however, was still PCI-DSS compliant, as it is only when data is transmitted from the POS terminal to other devices in the network system, that the data must be encrypted, in order to maintain compliance. Table 4.20 shows the authorization escalation risk attribute identification population for the risk taxonomy attribute entities.

***Table 4.20: Authorization Escalation Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Access Control | Authorization |
| Technical | System Protection - Physical | RAM |
| Technical | System Protection – Cyber Security | Malware |
| Technical | Communications Protection | Encryption |
| Operational | Physical Protection | RAM |
| Operational | System Integrity | Authorization |
| Operational | Information Integrity | Encryption |
| Management | Security Assessments | Authorization |
| Management | System Acquisition | POS |
| Management | Services Acquisition | Data |

#### *4.4.4.2 Internal System Asset Breach*

The attackers then use the authorization gained from escalated passwords to breach and control several internal File Transfer Protocol (FTP) servers on the Target network system. The attackers were then able to exfiltrate the information gathered, by sending the data via the FTP servers to Russian FTP servers acting as receivers. It is estimated the attackers collected and transmitted 11 GB of information using the FTP transfers to the Russian-based servers (Shu, Tian, Ciambrone, & Yao, 2017). The data was sent at times that the system was expected to be busy, and the data was anticipated to become lost in amongst the large volumes of normal data transfers. Table 4.21 shows the internal system asset breach risk attribute identification population for the risk taxonomy attribute entities.

***Table 4.21: Internal System Asset Breach Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Access Control | Server |
| Technical | Event Logging | Analysis |
| Technical | System Protection – Cyber Security | Authorization |
| Technical | Communications Protection | Encryption |
| Operational | Audit and Accountability | Authorization |
| Operational | Configuration Management | Server |
| Operational | System Integrity | Authorization |
| Operational | Information Integrity | FTP |
| Management | Security Assessments | Server Authorization |
| Management | Planning | Scheduled Data Transfer |
| Management | Risk Assessment | Data Transfer |
| Management | System Acquisition | Server |
| Management | Services Acquisition | FTP |

### 4.4.5 Security Breach Notification

At the first occurrence of the malicious activity, personnel in Bangalore, India as part of the security operations network were notified by the malware detection system that potential malicious activity was being recorded from the network. The Bangalore team then informed security personnel in Minneapolis. No further action was taken or deemed necessary by the personnel in Minneapolis. Three days later, another malicious activity alert was raised but no action was taken again. The malware detection system also notified the security center personnel regarding suspicious data transfer activity during the FTP transfers. It was only upon the United States Department of Justice alerting Target about potential data breaches that investigative action was taken, and any serious network analysis was carried out. An independent security researcher had posted information regarding breaches of target network during this time. Table 4.22 shows the security breach notification risk attribute identification population for the risk taxonomy attribute entities.

***Table 4.22: Security Breach Notification Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Event Logging | Detection |
| Technical | System Protection - CyberSecurity | Detection<br>Alert |
| Operational | Awareness | Alert |
| Operational | Training | Process |
| Operational | Contingency Planning | Process |
| Operational | Incident Response | Procedure |
| Operational | System Integrity | Action |
| Operational | Information Integrity | Action |
| Management | Planning | Procedure |
| Management | Risk Assessment | Documentation<br>Procedure |
| Management | System Acquisition | Detection |
| Management | Services Acquisition | Alert |

### 4.4.6 US Senate Testimony

According to John Mulligan in his testimony to the United States Senate, Target had invested costly resources into security technology, processes and personnel. The resources included installing a large-scale defensive network in depth with layers of protection, along with data loss prevention tools (Committee on the Judiciary, 2014). Yet, within three days of initializing and internal investigation, an independent team of forensic investigators confirmed that the system had been infiltrated and large amounts of payment information had been captured and transmitted beyond Target’s network system. Table 4.23 shows the US Senate testimony risk attribute identification population for the risk taxonomy attribute entities.

***Table 4.23: US Senate Testimony Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection - Physical | Infrastructure |
| Technical | System Protection – Cyber Security | Software |
| Operational | Audit and Accountability | Processes |
| Operational | Awareness | Personnel |
| Operational | Training | Personnel |
| Operational | Contingency Planning | Audit |
| Operational | Incident Response | Procedure |
| Operational | System Integrity | Defense |
| Operational | Information Integrity | Data loss |
| Management | Certification | PCI DSS |
| Management | Accreditation | PCI DSS |
| Management | Security Assessments | Compliance |
| Management | Planning | Defense |
| Management | Risk Assessment | Network |

## 4.5 TEST CASE 3: TESLA AND VEHICLE AUTOMATON

The high-end Tesla vehicle accident has been selected for analysis in the Proof-of-Concept case study because the case can be considered high profile and has attracted a large amount of attention that is extensively analyzed by various parties, including the US National Transportation Safety Board (NTSB). Using advanced analysis techniques, including 3-D laser scanning technology, the NTSB investigators documented aspects of the accident, such as crash location, vehicle positioning and impact points. There was a large amount of data recovered from the Tesla vehicle. It was fitted with multiple electronic systems which transmitted and captured vehicle performance data, which added to the information analysed by the NTSB. The NTSB has released a report (NTSB HAR-17/02) of their investigation into the incident, and it is used extensively in the following Proof-of-Concept case study analysis (National Transportation Safety Board, 2017)

### 4.5.1 Analysis

While a Tesla Model S travelled on a US highway in Florida, a truck and semi-trailer made a left-hand turn moving across two eastbound travel lanes onto a local paved road. The Tesla vehicle struck and then went under the trailer unit of the truck. The car then travelled 100 meters and collided with a utility pole. The Tesla broke the pole and carried on a further 15 meters, rotating before coming to rest. The driver, who was alone in the vehicle was killed in the crash. The vehicle was under autonomous control at the time of the accident.

### 4.5.2 Physical and Environmental Conditions

The driving conditions were clear, with a dry road, during daylight hours, and with no sun strike or glare. The proprietary collision information downloaded from the Tesla vehicle in the Tesla laboratories, showed that the vehicle was under autonomous control using the Traffic Aware Cruise Control (TACC) as well as the Autosteer lane-keeping function. These applications are part of the 'Autopilot' suite, which is the branded name used by Tesla to market their autonomous vehicle driver assistance programs. The vehicle was provided with automated emergency braking and forward collision warning systems, which were active at the time of the accident, but which failed to apply. The Tesla recorded data indicated the vehicle was travelling at 74 mph (119 km/h), however the posted highway limit in that area was 65mph (104.6 km/h). The truck was travelling slowly whilst making the left turn across the highway in front of the Tesla, which then passed under the semi-trailer unit. The driver, who was restrained by a seatbelt suffered blunt force trauma to the head, which was fatal. This was due to the roof structure of the Tesla being displaced to the rear, from contact with the semi-trailer. The safety devices normally associated with a modern vehicle, namely the airbags, failed to deploy when the car collided with the semi-trailer. However, the airbags did deploy when the vehicle collided with the utility pole, approximately 8.4 seconds after the collision with the truck. Table 4.24 shows the physical and environmental conditions risk attribute identification population for the risk taxonomy attribute entities.

***Table 4.24: Physical and Environmental Conditions Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection - Physical | Driving Conditions |
| Technical | Event Logging | Proprietary |
| Operational | Awareness | Driver |
| Operational | Configuration Management | Autonomous Control |
| Operational | Contingency Planning | Driver |
| Management | Risk Assessment | Driver |
| Management | Planning | Driver |

### 4.5.3 US National Transportation Safety Board Report

The US National Transportation Safety Board (NTSB) released a completed report that comprised a presentation of the factual data of the crash, vehicle damage, and driver information. Factors such as highway characteristics and regulation and policy are also listed. The NTSB defect investigation lists functionality defects identified, if any, in the provision of factual information. The salient points, which are relevant to this section, are that there were no mechanical failures of the Tesla vehicle determined by the investigation and that the manual braking system was fully functional. The NTSB report then presents a definition and description of the automated vehicle control system in use at the time of the crash. The report states that the supplemental or fully autonomous, full control driving tasks may be considered convenience systems, and that these systems are not designed to take over the control of the vehicle from the driver.

There are risk indicators discussed in the report, but also confusion of definitions of terminology that have originated from a sales and marketing domain, rather than an engineering perspective. Here is a cause for safety and risk specification concerns. The Tesla S model is marketed with several vehicle control systems that are promoted as being fully automatic. Traffic Aware Cruise Control (TACC), Autosteer, and Auto Lane Change (ALC) are functions that are marketed as part of the Autopilot suite. Other functions that are marketed as automatic are Forward Collision Warning (FCW) and Automatic Emergency Breaking (AEB) as part of a Forward Collision Avoidance (FCA) suite. Autopark, Speed Assist and Lane Assist are part of the crash avoidance and safety features of the vehicle. Table 4.25 shows the NTSB report risk attribute identification population for the risk taxonomy attribute entities.

***Table 4.25: NTSB Report Risk Attribute***

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection - Physical | Mechanical |
| Operational | Audit and Accountability | Mechanical |
| Operational | Awareness | Driver |
| Operational | Configuration Management | Autonomous |
| Operational | Training | Driver |
| Management | Risk Assessment | Driver |

### 4.5.4 NTSB Report Autopilot Determination

The NTSB HAR-17/02 report simplifies the descriptions of functionality of the two main components of the Autopilot suite as: Traffic Aware Cruise Control (TACC) that provides control longitudinally through acceleration and deceleration to maintain following distance. In addition, Autosteer provides control laterally through steering to maintain position within the travelling lane. Thus, the vehicle, when Autopilot is activated, can be classified as and SAE International Level 2 automated vehicle system. The SAE driving automation levels determine that, whilst both lateral and longitudinal vehicle control can be sustained, it is expected that the driver will achieve the minimal risk condition as needed. Thus the driver must be continually supervising the driving task, be engaged at all times and monitoring the environment and local conditions, which includes other vehicles occupying the local area, as well as pedestrians. Therefore, the driver at all times should be ready to intervene as necessary to maintain safe operation of the vehicle. Table 4.26 shows the NTSB report autopilot determination risk attribute identification population for the risk taxonomy attribute entities.

*Table 4.26: NTSB Report Autopilot Determination Risk Attribute*

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | System Protection - Physical | Autonomous |
| Operational | Awareness | Driver<br>Autonomous |
| Operational | Training | Driver |
| Operational | Configuration Management | Autonomous |
| Operational | Contingency Planning | Driver |
| Management | Planning | Autonomous |
| Management | Risk Assessment | Autonomous |

### 4.5.5 Autopilot Constraints and Warnings

The Tesla vehicle places both hard and soft constraints for a driver. The hard constraints are imposed automatically by the system, and the soft ones are cautions directed at the driver, either through the instrument screen or as instructions in the owner's manual. These include the use of the Autopilot function. The hard constraints include a set upper limit on lateral acceleration, cruise speed limited in accordance with speed limits (dependant on speed limit detection) and measurement of the driver's level of engagement. The engagement level of the driver is determined through the driver's interaction with the steering wheel through monitoring changes in the torque applied. There is a chain of increasing levels of warning. Designed to encourage the driver to interact with the steering wheel, starting with visual warnings, moving onto audible warnings which increase in volume and a final visual warning. If the driver does not respond to this final visual warning the vehicle will then slow to a stop in the current travel lane to a complete stop and activate the hazard flashing indicators. Table 4.27 shows the autopilot constraints and warnings risk attribute identification population for the risk taxonomy attribute entities.

*Table 4.27: Autopilot Constraints and Warning Attribute*

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Identification | Engagement |
| Technical | Authentication | Interaction |
| Technical | Event Logging | System |
| Operational | Awareness | Driver<br>System |
| Operational | Personnel Security | Driver |
| Management | Security Assessments | Driver<br>System |
| Management | Planning | Driver |
| Management | Risk Assessment | System |

### 4.5.6 Driver Autopilot Engagement

The driver of the vehicle engaged the autopilot system on four occasions during the 31 minutes leading up to the incident. This autonomous control was continuously engaged for the final 18 minutes and 55 seconds of the car's activities, until the vehicle crashed. Within 3 minutes of the final 18 minutes and 55 seconds, (more than 15 minutes before the incident) an auditory alert along with two visual alerts occurred, informing the driver that he was required to place his hands on the steering wheel. During the final 60 seconds, the driver's hands were only detected upon the steering wheel for 34 seconds. It is not stated which 34 of the final 60 seconds the driver's hand maintained contact with the steering wheel, although the NTSB report clearly states that there was no contact for the final 6 seconds. Three seconds before the incident, and up until the time of impact, the vehicle accelerated from 100 to 114km/h, an estimated acceleration of 5km/s. There was no pre-crash breaking, and no evasive steering maneuvers detected. Table 4.28 shows the driver autopilot engagement risk attribute identification population for the risk taxonomy attribute entities.

*Table 4.28: Driver Autopilot Engagement Risk Attribute*

| Domain | Family | Risk Attribute |
|---|---|---|
| Technical | Identification | Alerts |
| Technical | Authentication | Driver<br>Engagement |
| Technical | Event Logging | System |
| Operational | Awareness | Driver |
| Operational | Training | Driver |
| Management | Planning | Driver |
| Management | Risk Assessment | Autonomous |

## 4.6 CONCLUSION

Semantic analysis of the selected NIST best practice guideline provided the vocabulary input that was used for the taxonomy creation process of the three case studies taken from different IoT ecosystems. The output from the taxonomy creation process provided the information that allowed the researcher to fully populate the completed taxonomy entity structure. Thus, the Proof-of-Concept is shown, through manually performing the semantic analysis process steps. The taxonomy creation process shown in Figure 4.1 demonstrated the completion of a risk identification taxonomy for the pilot study.

The Security field was selected as the ‘Identifier’ of the taxonomical structure, which is the determination of the taxonomy subject area. The Identifier was then used to select NIST SP800-53 and FIPS 200 as a source for the vocabulary input. The application of the vocabulary input, within the structure of the selected case studies, identified three, top level taxonomy domains: Technical, Operational and Management. Each of these domains show several Family identifications when parsed through the taxonomy creation process. The Family entities were then used to provide Risk Attributes for each Family entity upon analyzing the three Case Studies and utilizing the vocabulary input. The risk attribute identifiers where then used to complete the taxonomy entity population. Thus, the manual taxonomy creation process performed provide the Proof-of-Concept for taxonomy creation. Chapter four demonstrates the steps and the effectiveness of the taxonomical structure creation process. The resultant Risk Attributes demonstrated knowledge and insight into risk evaluation and enumeration.

The Test Case Pilot study provides proof that a Taxonomical Creation Tool can be manually applied to create applied risk evaluation knowledge. In Chapter five the researcher will show further taxonomical development, through the integration of a Semantic Analysis Engine (SAE). The SAE will provide the ability to automate the Taxonomical Creation tool. The Semantic Engine will be applied to determine risk attributes by automatically analyzing semantic content provided by the Information Link Input. Thus, Chapter five will demonstrate how the engine will apply the resultant semantic content to real-world scenarios. The information thus provided will give insightful automated risk analysis.

# Chapter 5
# ADDRESSING BUILDING ISSUES

## 5.0 INTRODUCTION

Chapter five addresses challenges encountered while automating the manual taxonomy transactions of Chapter four. These are building issues and the need to consult with other IT experts in order to develop a functional and working solution. The Design Science Research Methodology (DSRM) adopted for this research informs the improvement of an artefact through expert input. The second artefact proposed in Chapter three is the development of an instantiation in the form of a process: The Semantic Analysis Engine (SAE). The input-process-output model suggests that the process itself can be varied and provide an output that will better inform the next stage of Taxonomic creation. The first artefact delivered, in the form of an inference, successfully provided the text data Information accumulation. The expert opinion adopted improves the artefact by forming a mature instantiation. The process of the DSRM expert input iteration is described in Chapter five and explains each of the development steps.

Chapter five is structured by the sequence moving from the primary manual evaluation of the instantiation artefact, the discussion of the proposed artefact maturity improvement, and the development of the process variations adopted to deliver the automated instantiation artefact. Section 5.1 describes the primary failure identified in the proposed method. Section 5.2 explains the successful output of stage one, with the text data information accumulation. Section 5.3 discusses the proposed stage two process hardware resource utilization estimation.

## 5.1 T-SQL SEMANTIC ENGINE PRIMARY FAILURE

The primary purpose of the proposed Phase two, as identified in Section 3.3.2, is to produce a Semantic Analysis Engine (SAE). The SAE output is to be used to inform the Maturity Model through text analysis providing automated taxonomical information. The process proposed, was designed to produce the required output in the form of an information vocabulary following a process of text mining, as described in Section 3.1.2, step 3. The proposed SQL sub-processes were designed to produce a pre-processed set of data, ready for text analysis to provide input for the following phase: Taxonomy Creation. The initial installation of the SQL database was straight-forward and presented no difficulties. The database was installed on a dedicated hardware platform of the specifications shown in table 5.1.

***Table 5.1: Test Specifications***

| | |
|---|---|
| CPU | Ryzen 5 3400G |
| RAM | 32GB |
| MAIN BOARD | ASUSTek Prime B350-PLUS |
| VIDEO | NVIDIA GeForce GTX 660 Ti |

T-SQL software was selected in the initial stages of this research after consideration and expert information input, for several reasons: The researcher is qualified in the setup, configuration and administration of the product. The product has extensive literature available that indicates that the product is fit for purpose. The product is a commercial product, is very mature, and is in use as a respected business product. The product is available as part of the university wide license and is thus free for the researcher to use for the duration of this research experiment.

During the process of applying the method that was proposed in Section 3.1.2 it became apparent that there were difficulties when applying the core modifications to the closed source software required to produce a Semantic Analysis Engine. One of the difficulties of using closed source software is that the software contains closely guarded proprietary code. This can be view as leasing the right to use software, rather than purchasing the software. It was expected that full support of the product functionality and constant innovation would provide a better product for this research when using closed source software. However, in the setup and configuration of the experiment for the research, the proprietary closed source software did not perform to expectations. Upon further investigation, there were limited informational resources available. Even though the software was marketed to have full capabilities required for this research, the information was found to be erroneous.

When investigating alternatives, in the form of open-source solutions there are normally several challenges presented to the researcher. These are quality of support, usability, service and availability. Other open-source considerations such as security and innovation did not form part of the decision-making process when determining utilization of product software. An open-source solution was then suggested. The open-source software that was identified to provide the best fit for this research was found to be mature, fully supported, and provided the level of usability required to perform Semantic Analysis Engine processes.

The utility of adopting Design Science (DS) as the overarching research method became clear when presented with the challenges described above. As shown in Figure 3.2 and described in Section 3.2.1, there is a step from theoretical design and to application functionality. This gap pertains specifically to the Semantic Engine design, construction and integration. The research design identified the artefact design and development stage would be the pivotal phase of the research process sequence. The research design recognized that expert input would be integrated into this stage, as part of a Design Science expert evaluation and selection input process iteration. The modular Design Science approach to Information Systems adopted as the research design, provided opportunity to the researcher to provide a variation to the Semantic Engine construction that would not affect the following research process sequences. The research design identified the Semantic Engine as an artefact in the form of an Instantiation as shown in Figure 3.3 and described in Section 3.2.1. The ideal of DS research is to produce artefacts that have undertaken a process of improvement through a process of iterative expert input. After removing the emphasis on a particular software product, the underlying requirement for this instantiation artefact was to produce a system implementation prototype designed to automate and provide the input, to inform the Taxonomy generation process. Therefore, the utilization of an open-source software solution fits within the research design parameters, demonstrating the efficacy of research methods developed through DS adoption.

The following sections discuss the initial method applied, the problems encountered, and the variations for the Semantic Analysis Engine (SAE) application. First the text selection process input, and information accumulation steps will be addressed. Then the variations to the domain relevant term extraction component of the Semantic Engine Process will be discussed. Finally, the alterations to the Semantic Analysis Engine will be presented step-by-step, to provide a new sequence from information accumulation input, identifying the processes applied, and the output design to inform the taxonomy creation process.

## 5.2 INFORMATION ACCUMULATION PROCESS

The identifier and the information linking requirements for the text input selection process was performed as expected and described in Section 3.1.2 and is shown in Figure 5.1. A selection of 20 peer reviewed papers were taken from the IEEE transactions on IOT in keeping with the text input selection process parameters. This is depicted as the top-level Database Text component of the original Semantic Engine process as shown in Figure 5.2.

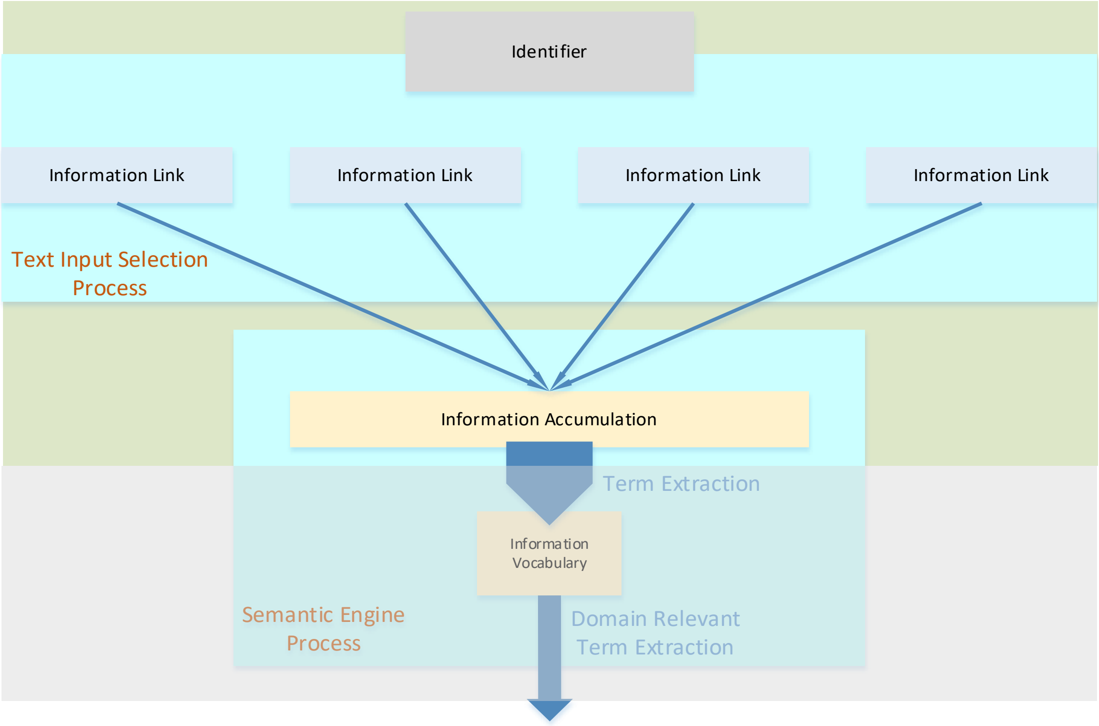


***Figure 5.1 Successful processes***

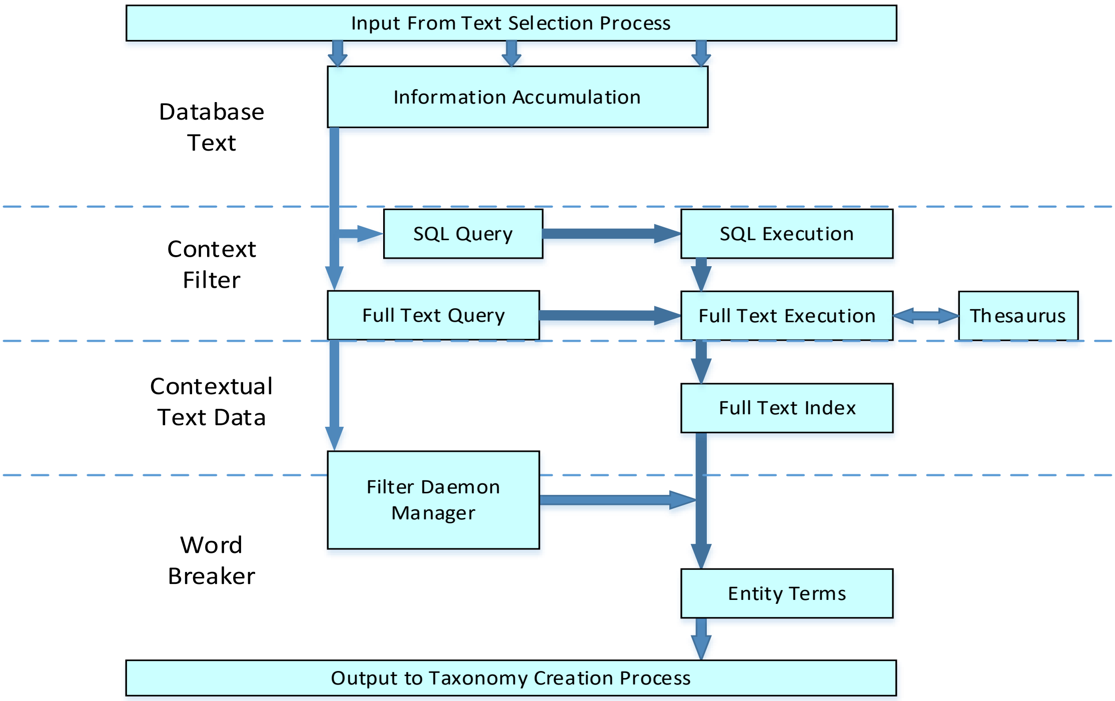


***Figure 5.2: Original Semantic Engine process***

## 5.3 EXPECTED RESOURCE UTILIZATION

The Semantic Engine process is divided into four logical process steps. The first step is creating the textual database, where a body of selected text is input into the SQL database. The body of text created is an accumulation of information which is output from the first of the process steps, which then informs the second step of the process, text mining. The first stage was projected to utilize under 25% of the system resources. The database text in the raw format is then to be subject to four SQL intensive sub processes as part of the second step of the process. There is also a comparator section of the second stage where the SQL output are compared to a selected thesaurus. The second stage was identified to utilize the highest level of computing resources. The second stage was projected to use 60% of the resources. The third section processes the information from the intensive SQL query processing output from step two, to create a full text index. The purpose of the third step is to take the output and determine contextual text data, providing the full text index. A parallel process was to be undertaken in the third step. The sections shown in Figure 5.3, depict the proposed process blocks to be implemented within the SQL based semantic engine and the relevant resource utilization prediction.

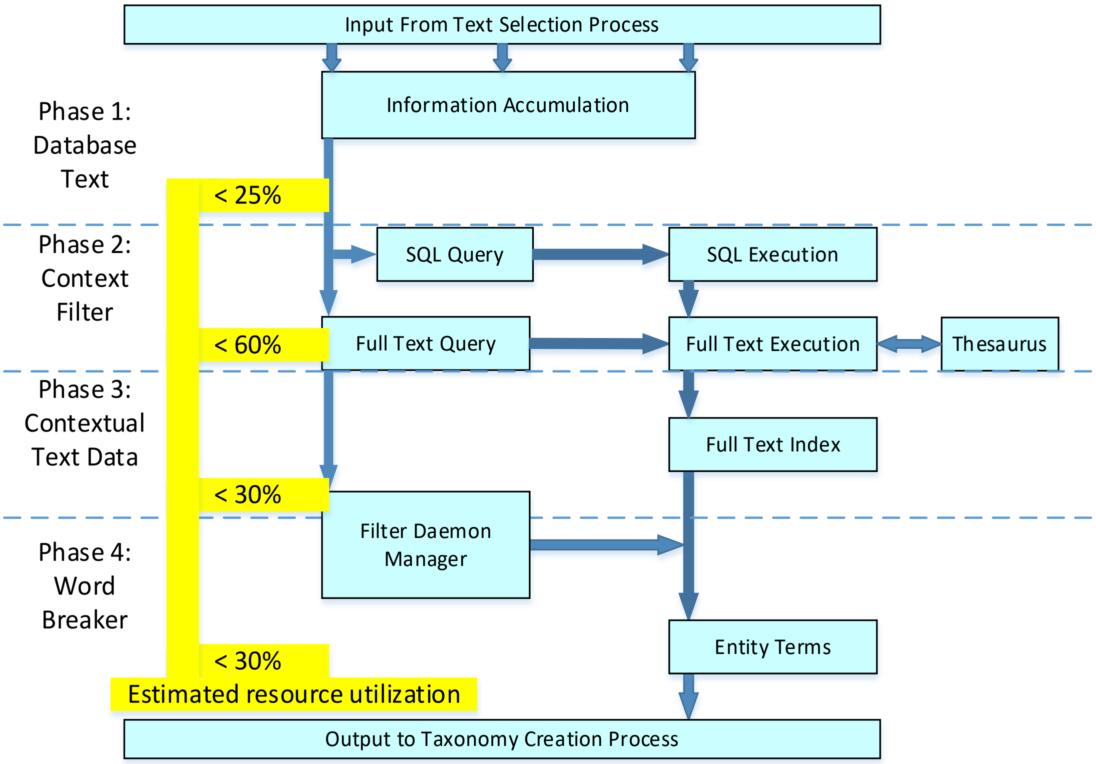


*Figure 5.3: Expected resource utilization*

## 5.4 OBSERVED RESOURCE UTILISATION

An initial installation of the database on the hardware platform was determined to be normal and provided standard functionality. Identifiers from the information link defined in Chapter three, determined text as part of the text input selection process. From this information the database was then populated with (20) papers. The process followed the predetermined protocol determined to produce a library of text, ready for semantic analysis. The database demonstrated initial functionality, as the text was able to be parsed and the title searched. This provided primary functionality requirements for text mining. Therefore phase 1 of this process, Database Text was determined to be complete and functional.

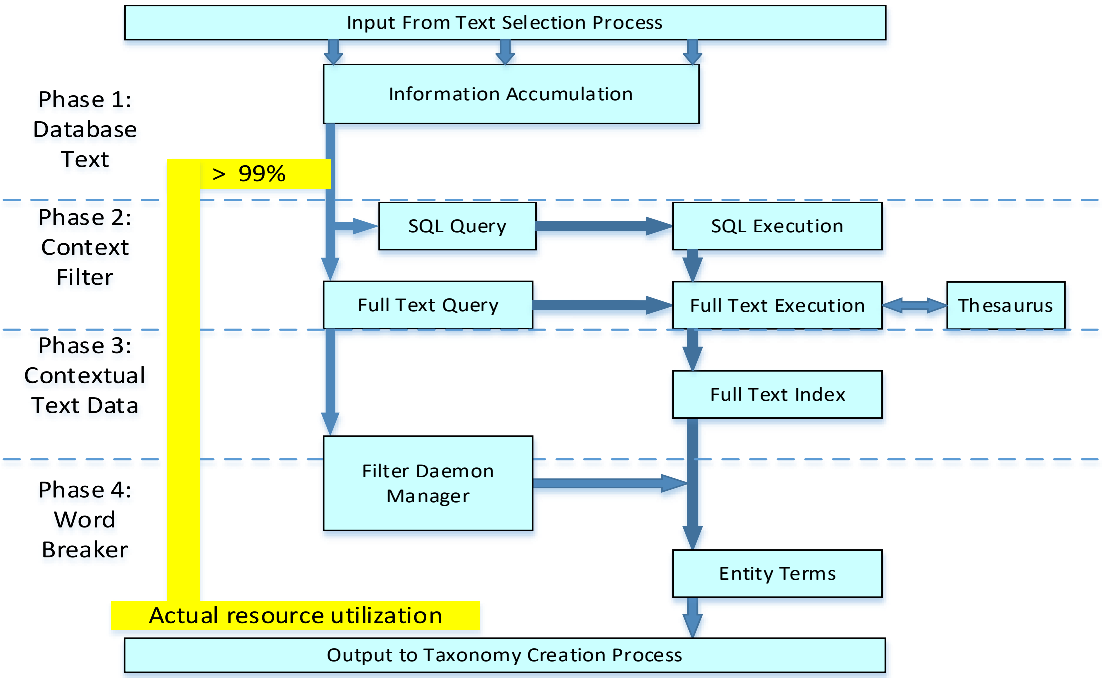


*Figure 5.4: Actual resource utilization*

However, the following phases of functionality were not reached, and implementation was problematic as the resource usage was continuously at levels greater than 99%, as shown in Figure 5.4, and did not allow any but basic processing functionality. The implementation process was difficult and the relevant support literature regarding the setup and administration was either hard to find, or lacking configuration/load balancing details. The overarching requirements of the database for this research are flexibility and adaptability, which required a level of configuration that goes beyond the basics of textual searching and parsing. There were few information resources available other than vendor courses which were held in locations that were geographically difficult to access, and expensive.

The resource components of the hardware configuration, designed for the Semantic Analysis Engine were expected to provide ample resources for the proposed SQL analysis. Of the resource components, Hard Drive and CPU utilization were determined to be important indications of functionality. The Hard Drive, although a platter drive design, was designed to be used in an enterprise business environment and as a storage area network device where multiple read/writes are expected. The CPU installed provided four CPU cores allowing eight hyper-threads along with 11 GPU cores. Figure 5.5 shows the observed disk resource utilization and Figure 5.6 depicts the observed CPU resource utilization. The two figures show the actual utilization of these two important resources when populating the Information Accumulation Text Database. These two figures indicate that further SQL processing would be problematic or impossible.

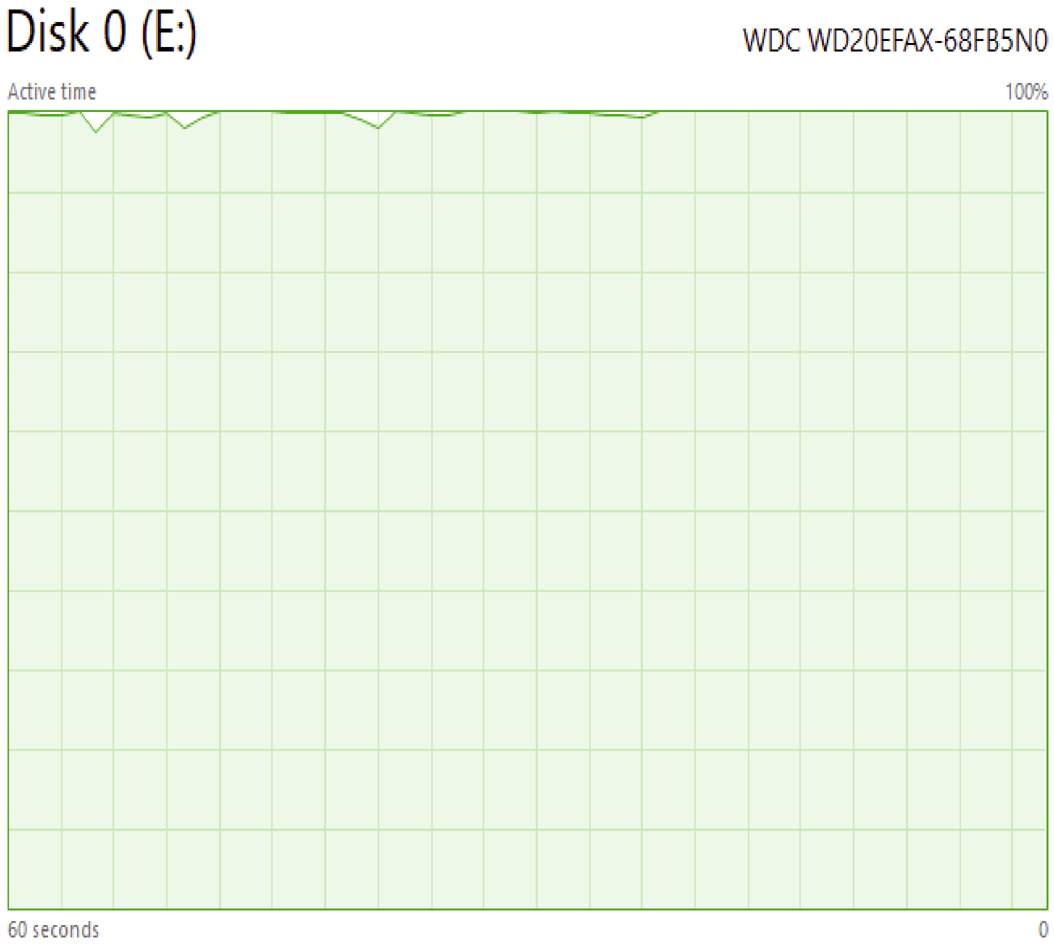


*Figure 5.5 Disk resource utilisation*

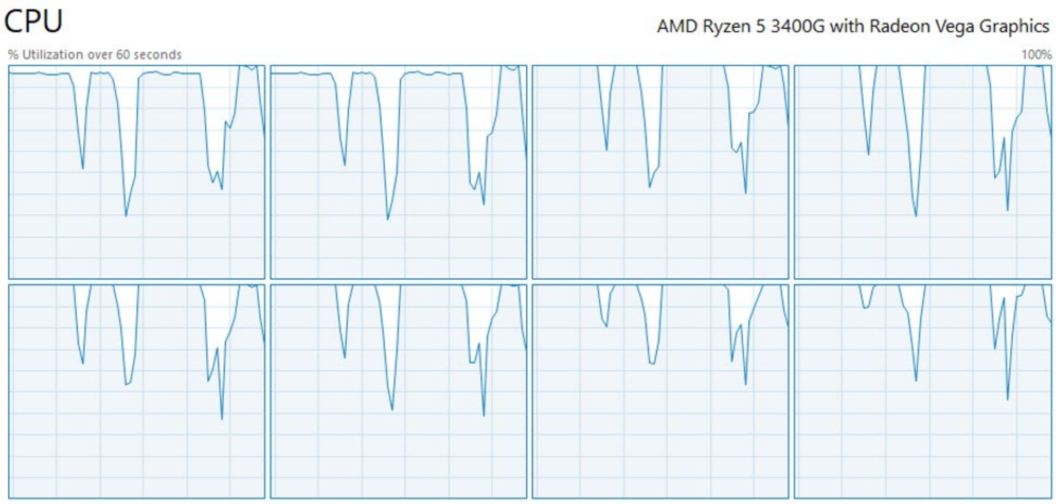

CPU
% Utilization over 60 seconds
AMD Ryzen 5 3400G with Radeon Vega Graphics
100%

*Figure 5.6: CPU resource utilisation*

## 5.5 INFORMATION ACCUMULATION FROM DATABASE TEXT

Considering the four SQL components proposed: Database Text, Context Filter, Contextual Text Data, and Word Breaking (see Figure 5.2), the Information Accumulation data is of most concern.

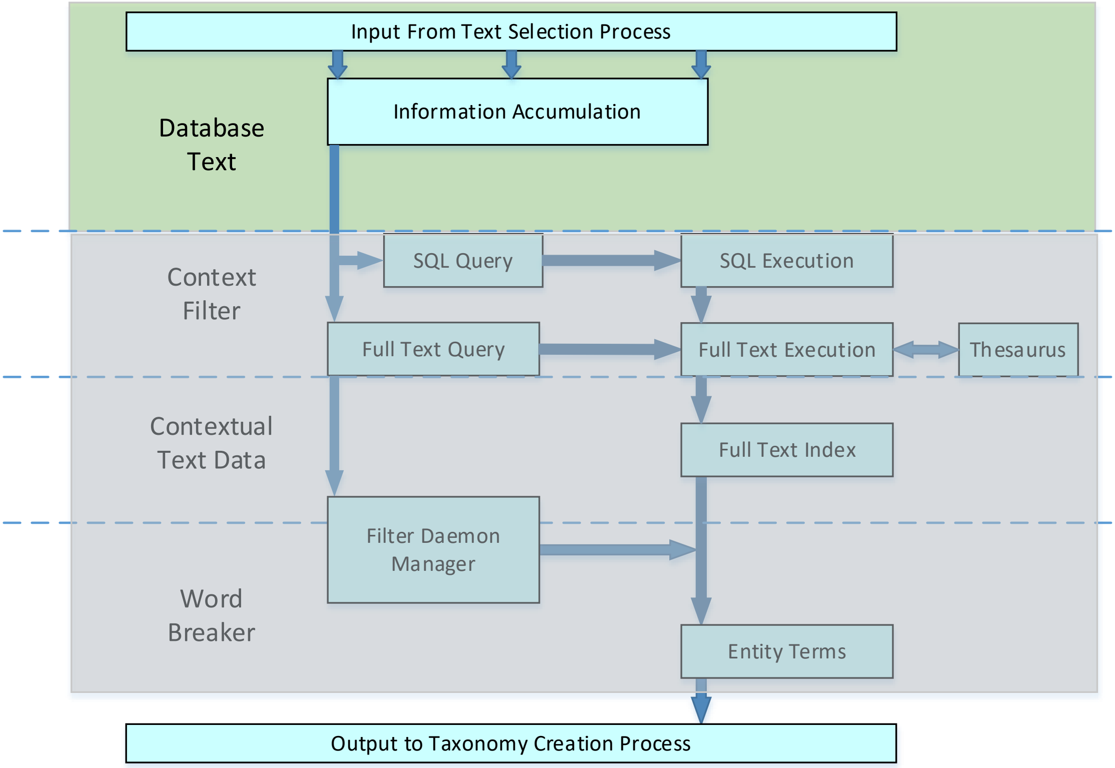


*Figure 5.7: Successful information accumulation*

Simple word level features and simple models pertaining to large amounts of Information Accumulation data will provide improved results when compared to a feature rich sophisticated model with less data. As expected, and depicted in Figure 5.3, resource utilization with sophisticated models and deep feature extraction with SQL tools, are the most computationally intensive. The observed utilization of resources were unreasonably high, allowing only the import of text, and no algorithmic computation was possible. Therefore, after having created a large volume of text-based data, the processing of the data from the information accumulation was not possible. As shown in Figure 5.7 the algorithmic computational sections of Context Filtering, Context Text Data identification, and Word Breaking have been reassessed to provide a working and achievable result, and thus have been ‘blanked out’.

## 5.6 MODIFICATION TO SPECIFICATIONS

As part of the Design Science methodology, an information input loop was performed, with expert opinion sought from several sources. The input was designed to assess the database assembly as an artefact in terms of functionality and implementation. The input from a technical expert regarding the database hardware platform, informed the researcher that there had been problems observed by other researchers. When the database was applied and researchers were attempting to customize the platform, similar hardware utilization problems occurred. Input from a software programming expert was then sought. The expert opinion indicated that using ‘closed source’ software would produce the difficulties regarding customization that was observed to be problematic. The problems that were presented regarding information and literature resources were determined to be part of the nature of utilizing proprietary vendor, closed source code.

The researcher integrated this information into a further Design Science feedback loop. As part of the artefact modification process, an investigation into open-source platform availability was carried out. The investigation determined that open-source platforms provided resources that would produce a more suitable artefact. Therefore, three components, Context Filter, Contextual Text Data and Word Breaker, depicted in Figure 5.8, specification modification below, indicated logical process steps that required modification. Thus, an open-source solution has been investigated to provide a plug-in modular capability, to take information from the database text or information accumulation and provide a processed output of entity terms to inform the taxonomy creation process.

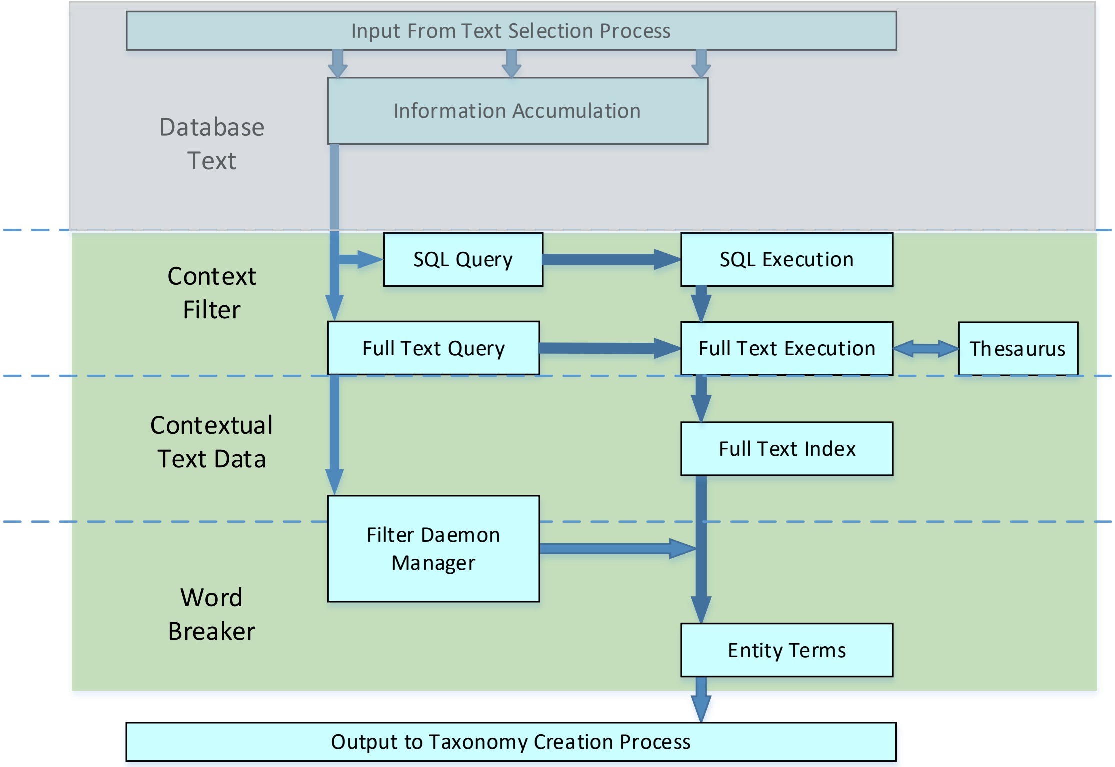


*Figure 5.8 Specification modification*

## 5.7 SUITABILITY REQUIREMENTS:

As described in Section 3.2 there are requirements necessary to develop a functioning Semantic Engine. There are three main components required to achieve this goal: a collection or grouping of text data, an analysis of the representations of the data, and some method for capturing regularities within the data. The representation of the first component, data grouping, is labelled the Information Accumulation within this research, when utilizing this form of semantic analysis. Aspects of the representations of the Information Accumulation text data are called features, which may be superficial and easy to extract, such as the words and sequences of words themselves, or deep and more difficult to extract, such as the grammatical relationship between words. This requires that algorithms or models are applied to the Information Accumulation to identify and enumerate regularities in the data in terms of the extracted features to determine output for this research.

Although SQL databases are designed for large data sets of up to 2 GB of information per cell which can come in a variety of forms, this research is primarily concerned with processing large amounts of data in the form of text, with a view to produce other types of data such as relational and graph data. The ideal solution required to provide the output to this research falls into the disciplinary boundaries of Natural

Language Processing, which is designed to assist with textual language information determination.

## 5.8 NATURAL LANGUAGE PROCESSING

In the simplest terms Natural Language Processing can be defined as computer manipulation of natural language. Natural Language Processing provides a researcher tools to investigate information contained within unstructured text. Natural Language Processing has a wide scope and many applications. It is accessible to the researcher as part of a mature, open-source development called the Natural Language Tool Kit (NLTK). The NLTK has been developed since its initial release in 2001 by Stephen Bird and Eduard Roper from the Department of Computer and Information Science at the University of Pennsylvania. The information and support for the tool kit is extensive and easy to locate. It is based and written in Python programming language which is also open source with extensive information and resources. The NLTK has been designed as a teaching tool and platform for building research systems and prototyping, and thus is ideally suited for the proposed research tasks. The following sections will define the steps of the process that will be undertaken to provide a plug-in to replace the SQL components of the Semantic Analysis Engine. In the block diagram in Figure 5.9, the NLTK will provide the necessary steps of term extraction information vocabulary and domain relevant term extraction.

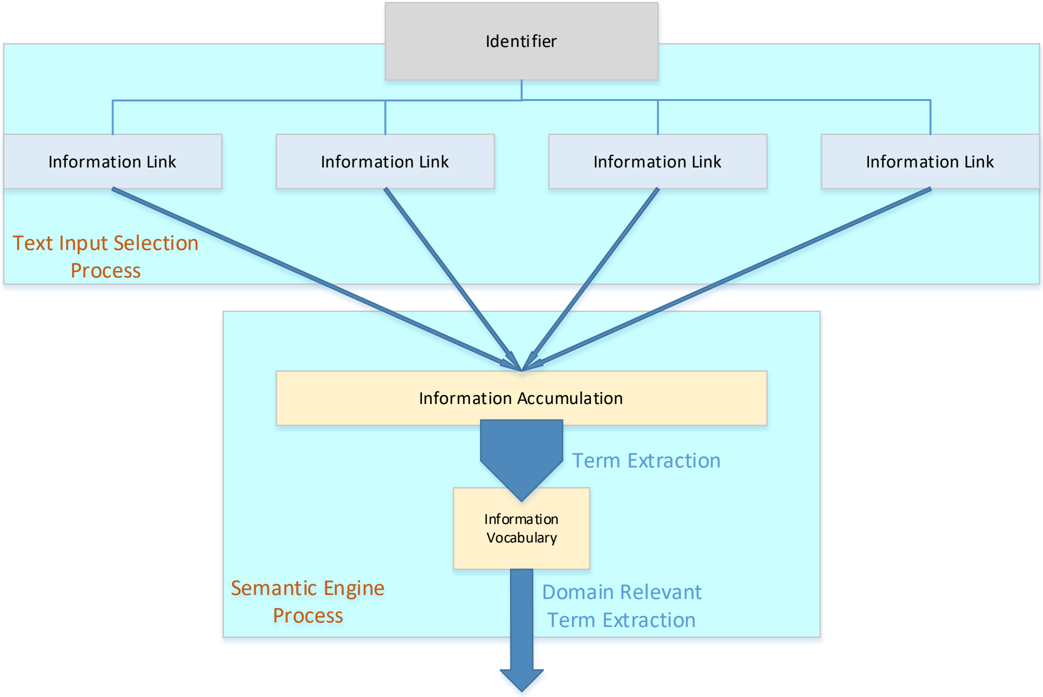


*Figure 5.9 Process block diagram*

## 5.9 RELEVANT TERM EXTRACTION SUB-PROCESSES

Several sub-process steps have been identified to utilize the input from the established information accumulation of text data, to provide domain relevant term extraction to inform the taxonomy creation process. These sub-processes are normalization, word breaking/tokenization, tagging, chunking, stop word removal, and finally, text classification. Each of the sub-processes used in this research are discussed in the following sections.

### 5.9.1 Normalization

Text normalization is a solution for overcoming or reducing linguistic noise, which consists of information that is not useful. The normalization task can be approached in two stages: first, the identification of errors in an input text, and second, lemmatization, or the reduction of inflectional endings to a base form of the word. A primary normalization approach is to refer to a text of known correctly spelled, relevant words to detect in-vocabulary (IV) and out-of-vocabulary (OOV) terms, with respect to the known text. For the second stage, a word list of base words, and the inflectional endings is referenced. An example of the lemmatization process is the word configure which is the normalization of the words: configured, configuring, configuration, configures, and configurable. A more complex structure may separate the five words in the example into two subsets. The first subset contains the words configured, configuring and configures in the second subset contains the words configuration and configurable, which may be of value within this research. The normalization process can be basic or more advanced. Basic normalization deals with errors detected, and provides lemmatization efficiency before the tagging stage, by removing unknown words that are not relevant, replacing misspelled words, and providing the base word which may be used to condition the output parameters.

Most practical applications leverage the simpler approach of replacing non-standard words with their standard counterparts as a "one size fits all" task. Baldwin and Li (2016) devised a method that uses a taxonomy of normalization edits. These researchers have evaluated this method on three different applications: dependency parsing, named entity recognition, and text-to-speech synthesis. The method categorizes edits at several levels of granularity and the results demonstrate that for this research, a taxonomy is an efficient approach to normalization.

Advanced normalization is more flexible, taking a lightly supervised automatic approach trained on an external dataset which provides annotation with short forms linked to the equivalent long or corrected forms. This can be particularly useful in this research when determining terminology of a technical nature. Demir (2016) describes a method of context-tailored text normalization. The method considers contextual and lexical similarities between standard and non-standard words, which is again used to reduce noise or information that is of little use. The non-standard words in the input context in a given sentence are tailored into a direct match, when there are possible shared contexts. A combination of these approaches will be adopted to condition the output.

### 5.9.2 Word Breaking / Tokenization

Word Breaking or Tokenization is the first step in processing text that is human readable into components that can be analyzed through a computational process. The primary task in this stage, is to split the text into a series of words. To begin this task, it is necessary to separate the words from punctuation and other symbols. A tool that performs this word breaking process is called a tokenizer. In some languages, the question of what indicates a word is not simple. When performing text analysis, there are strings of characters that are clearly words, but there are also strings for which it is not clear. For most times during the word breaking process, punctuation needs to be separated from words and eliminated. However, some abbreviations might contain punctuation characters as part of the word. Dates, times, measurement unit figures, and monetary units are examples where removal of punctuation may cause analytical difficulties. When treating numbers, if they contain commas or dots, the numerals should not be separated, which allows the creation of a simple regular expression rule. However, difficulties arise when considering what to do with contractions such as "don't,' where it may not be useful to expand it into the two words "do" and "not."

Tokenization usually consists of two subtasks: sentence boundary detection and token boundary detection. For the purposes of this research, the first task provided sentence boundaries, and the second task is to detect token boundaries. Horsmann and Zesch (2016) evaluated a method for dealing with token boundaries consisting of three steps. First, the researchers split the text according to the white space characters. Then they employed regular expressions to refine the splitting of alpha-numerical text segments from punctuation characters in special character sequences. Finally, these sequences of punctuation are reassembled. The process is to merge the most common combinations of characters into a single token. However, the use of word lists will be created that merge abbreviations with their following dot character, in this research. White space is a good indicator of separate words, and for the purposes of this research, which is based upon the English language, white space identification will be used to determine word separation. However, any missing whitespace characters may present a challenge to the task of tokenization. The process identified of using white space boundaries will increase output accuracy.

### 5.9.3 Tagging

Tagging is the process of explaining how a word is used in a sentence as a Part of Speech. For the purposes of this research, Parts of Speech are broken down to seven main parts being: Noun, Pronoun, Adjective, Verb, Adverb, Preposition, and Conjunction. Examples of each these are given below. Whilst Parts of Speech can be divided into subclasses, a tagging process labels seven classifications. An example of this process is that formal nouns are determined through the use of a capitalized first letter. The classification process described does however present complexities, for example in the last example, all sentences start a capital letter. Therefore, structural processing is required at this stage. The seven Parts of Speech utilized within this research are identified as:

- Noun (N) John, Auckland, Vehicle, pump, computer
- Verb (V) drive, control, change, redirect, process
- Adjective, (ADJ,) large, operational, initial, final
- Adverb (ADV) quickly, well, always, never
- Preposition (P) in, at, with, about
- Conjunction (CON) and, or, because, since, if but
- Pronoun (PRO) I, you, she, he, but, they, we, me, us

### 5.9.4 Chunking

The process of analyzing unstructured text and extracting phrases is called chunking. This process takes place before tokenization. The general mechanism utilized to provide chunking utilizes a regular expression term to generate the appropriate output. The chunking process is performed after the Part of Speech tagging process. An example of this process is noun phrase chunking where the chunk grammar is identified through a single regular expression to provide a chunk grammar part of speech in tags. Libraries such as Spacy and TextBlob will also be used which give such phrases in the form of a text input. The regular expression rule is utilized to detect an Identified Determiner (ID), followed by one or more adjectives (ADJ) and then a noun (N). The process will then ascertain a chunk (C) which will be utilized in the categorization process. An example of the chunking process is shown in Figure 5.10. In this example, the sentence used is 'The quick brown fox jumps over the lazy dog.' Two ID's are located, providing initial divisions of: 'The quick brown fox' and 'The lazy dog' The words 'jumps (V)' and 'over' (ADV) are therefore not included in the chunk output. Thus, the two chunks identified, upon removal of the ID are 'Quick brown fox' and 'Lazy dog.' The sequence of tokens, in this case is the two words: 'jumps' and 'over' that are not identified as chunks. They are not disregarded but are determined to be a 'chink' and may provide additional value. In the given example, it can be shown that 'fox' jumps over 'dog' whereas it is not shown that 'dog' jumps over 'fox' demonstrating that the chink 'jumps over' does present additional value.

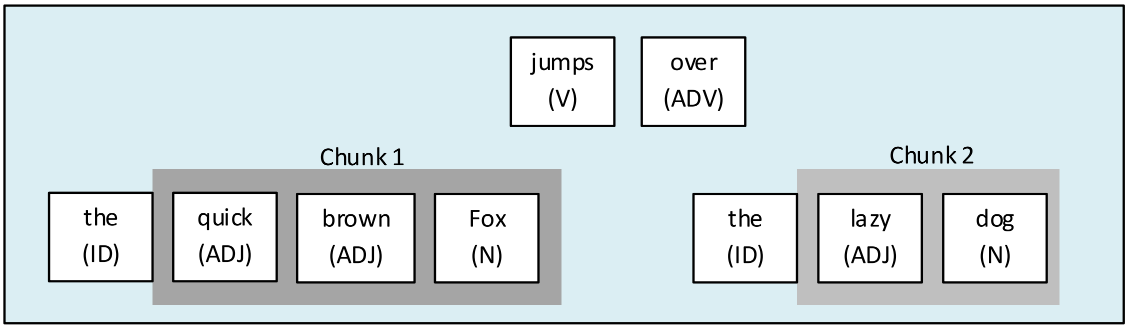


*Figure 5.10: Chunking example*

### 5.9.5 Stop Words

A step that increases the efficiency of the process, is to remove commonly used words (such as "if", "but", "and", "of") that the semantic engine has been programmed to ignore. These commonly used words are called stop words. They present problems, especially when indexing entries pre-processing and retrieving results upon the application of the algorithmic processes of the semantic engine. Stop words also take up space in the database and add to the computational overhead resulting in additional time to produce usable results.

In any natural language stop words are most common words utilized. Examples of these stop words are: if, and, but, then, which, and why. Stop words however do not add value to the meaning of documentation especially when analyzing text data and building NLP models such as required for this research. Therefore, a list of common words identified has been generated for the purpose of removing the stop words from the accumulated text data. The benefits of removing these stop words will help improve the performance as the tokenization process will only apply to meaningful words with the purpose of increasing classification accuracy.

### 5.9.6 Text Classification

An important phase of this research is Text Classification, where text is sorted into categories. There are difficulties in this part of the process when text is classified and then assigned levels of relative importance. Before classification levels can be assigned for identifying categories, assigning relative importance presents the researcher with a more complex task. The classification process involves ranking text according to criteria determining relevance and providing a scaled indicator. The classification process provides value to this research when assigning maturity capability levels and risk levels. Therefore, the output from the modified Semantic Analysis Engine is shown to inform the next stage of the research process, Taxonomy creation.

## 5.10 INTEGRATION OF METHOD VARIATION

The phases, processes and sub-processes outlined in Section 3.3 were designed to be modular, and therefore the variation to the method is integrated without disruption to the data process flow. The six NLTK sub-process steps described in Section 5.9 replace the six SQL sub-process steps contained within the original Semantic Analysis Engine process. Thus, there is no requirement for additional changes to the overarching method proposed for this research. The integration and sub-process steps are depicted in Figure 5.11: Semantic Engine variation.

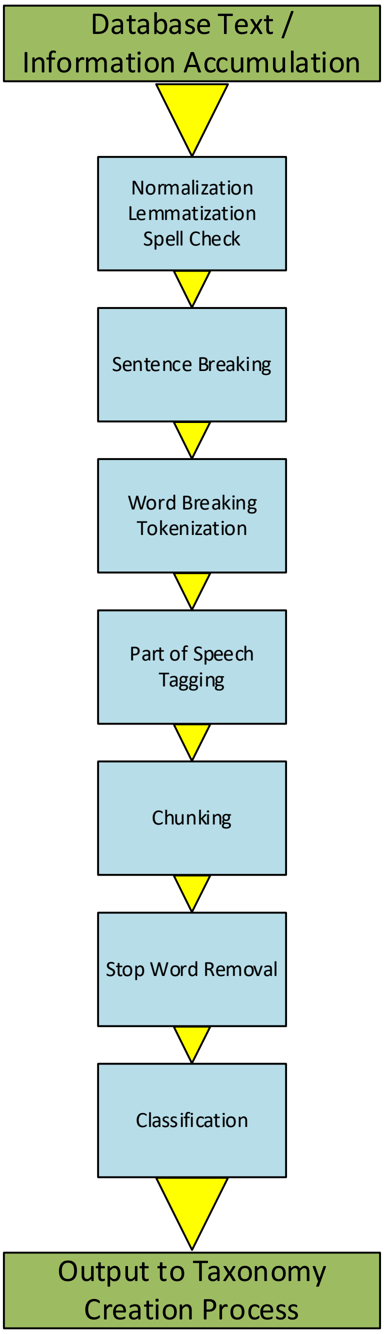


*Figure 5.11: Semantic Analysis Engine variation*

## 5.11 CORPORA AS DATABASE TEXT / INFORMATION ACCUMULATION

The Database Text / Information Accumulation generated during process step two, as shown in Figure 3.1 and described in Section 3.1.2, steps two and three, is utilized, however in a different format than that required by the original SQL method proposed. There is little modification required to adapt the Database Text into a corpus, ready for the sub-process steps identified in Section 5.9 and shown in Figure 5.11, SAE variation.

A Corpus is defined as a collection of text-based documentation, stored in a format that the NLTK recognizes. A Corpora is a collection of several Corpus and is thus the plural of Corpus. As the process of creating the Database Text already converts any PDF and Word publication format, the document text produced by the information accumulation is already in an NLTK recognized format. The conversion of the Database Text containing the 20 selected IEEE journal papers into an NLTK readable format created a Corpus of 69,819 words or 78 A4, 12-point, single spaced text pages. The creation and development of a specialized IoT Corpora of different, defined Corpus as an output of this method variation will provide a unique and distinct research contribution.

## 5.12 CONCLUSION

Chapter five addresses a challenge identified during the development of the Semantic Analysis Engine, as Phase 2 of the research design. The challenges presented involved resource utilization anomalies when setting up the hardware test system. A Design Science iteration was performed, and an artefact design improvement process was integrated into the research design. The implementation of an open-source software solution has been defined in Chapter five, to replace the original design utilization of proprietary software as the basis of the Semantic Analysis Engine platform. The integration of open-source software as a design improvement demonstrates the efficacy of the adopted Design Science Research Methodology as the overarching research strategy.

The method variation introduced is used and the results presented in Chapter six. These are the automated findings. Each of the sequential process steps outlined in Chapter six exhibits the application of the improved Design Science instantiation artefact. The three test cases outlined in Chapter four are each subjected to an automated analysis by the Semantic Engine implementation. The input command and the corresponding output of each of the process stages are reported.

# Chapter 6
# AUTOMATION FINDINGS

## 6.0 INTRODUCTION

Chapter five described the changes adopted as part of the Design Science Research Methodology (DSRM) implementation. An expert input iteration directed the research towards an open-source adaption to the Semantic Analysis Engine (SAE) as described in Chapter five to produce an improved artefact. Chapter six demonstrates the application of the improved artefact and presents the sequential process steps of the improved SAE artefact. Each of the three case studies identified in Chapter four, Maroochy, Target, and Tesla are processed through the SAE automation processes. Chapter six shows the findings derived from the output of each process step that informs the next process step in the SAE. The primary input is derived from the database text information accumulation identified as a result in Chapter four. Therefore, the summary of findings are presented in Chapter six, and demonstrate the output of each automated process stage, as identified in Section 3.1.2, and modified in Chapter five. The full findings and data sets are presented in Appendix C.

The three case studies are analyzed by the SAE individually in each of the following sections. Test case one, Maroochy Shire Sewage Spill is subjected to an automated analysis by the SAE in Section 6.1 and has been called ‘Maroochy’. Test case two, Target Private Information Data Breach is subjected to an automated analysis by the SAE in Section 6.2 and has been called ‘Target’. Test case three, Tesla Vehicle Automation is subjected to an automated analysis by the SAE in Section 6.3 and has been called ‘Tesla’. The three test cases are subjected to the same overarching process, in the same order but each generates different output data. Each section will follow the process described in Section 3.2 designed to extract domain relevant terms from the information accumulation. Section 3.1.2 described the SAE steps and sub-processes which are followed in the following sections. Each process step and sub-process is identified as ‘**In 6.x.x**’ to mark the input command. Any output is identified as ‘**Out 6.x.x’**, corresponding to the Input command identification. The Input commands are described, and the resultant Output identified (See Appendix C for full input and output data sets).

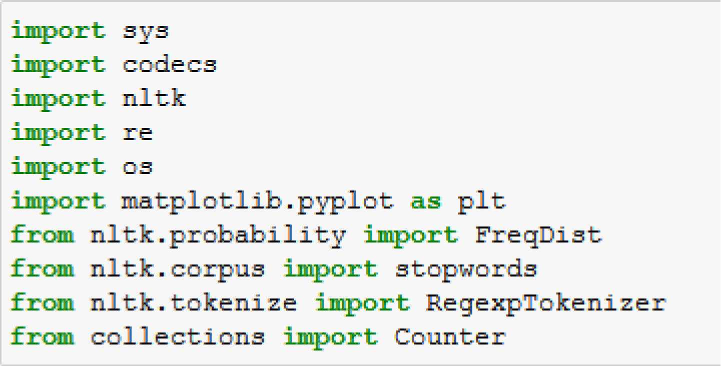
Thus, the three sections of Chapter six summarize the automation process steps, text selection, and filter management processes, in an input-process-output logic progression. The sections begin with an introduction overview describing the information accumulation and the corresponding database text for each case study. The three sections of Chapter six analyze information from three different IoT case study applications from diverse domains, as described in Chapter four. Each section of Chapter six is designed to utilize the same process steps, and output data from each step. The results and data will then be analyzed and discussed in Chapter seven, addressing and answering the research questions. (See Appendix C for the full Chapter six data sets).

## 6.1 MAROOCHY

The Maroochy Shire Sewage Spill (Maroochy) test case begins the SAE automated process. The Maroochy test case is the oldest of the three test cases, occurring in March 2000. The Information Vocabulary Text Database information includes information from many forums including Australian legal proceedings, Massachusetts Institute of Technology analysis publications, and textbook analysis of the subject. The forums provide the information links, supplying the Information Vocab, introduced as a corpus of 33,222 words. The corpora informs the SAE process throughout each process and sub-process steps. The results are subject to a word-frequency analysis. A Word-Cloud visualization is then applied, to inform further training steps. The final stage of the SAE process provides a domain relevant term extraction after a cluster analysis (See Appendix C.1 for the full Maroochy data sets).

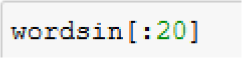
### 6.1.1 Process

In 6.1.1.1



```
import sys
import codecs
import nltk
import re
import os
import matplotlib.pyplot as plt
from nltk.probability import FreqDist
from nltk.corpus import stopwords
from nltk.tokenize import RegexpTokenizer
from collections import Counter
```

Input 6.1.1.1 shows the packages and modules that will be required to perform the various functions within the semantic analysis engine. The **sys** module is imported to provide the Python interpreter information regarding constants, functions and methods, when using the ‘dir(system)’ call. The **codecs** module defines the base classes for standard Python encoders and decoders (codecs). The **codecs** module allows the Python interpreter to encode text to bytes and also encode text to text and bytes to bytes. The error handling lookup process is also managed by the **codecs** module, and so on. Stopwords are imported from the **nltk.corpus** package of texts, providing a default list of **stopwords**, such as ‘and’, ‘if’, ‘then’ and ‘of’. Similarly, Input 6.1.1.5:

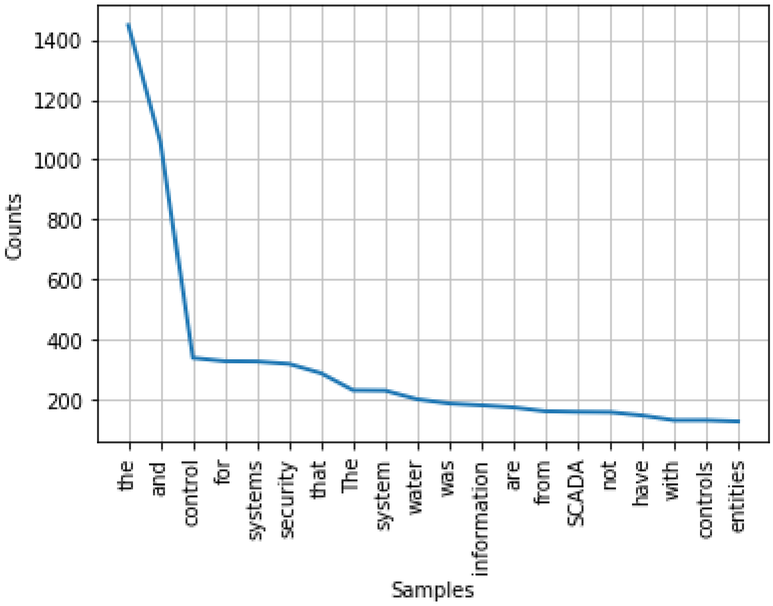


```
wordsin[:20]
```

; calls for the first 20 words from the tokenized stream labelled ‘wordsin’ The purpose of Input 6.1.1.5 is to confirm the word-breaking process has been implemented correctly, by retrieving the first 20 words of the Maroochy corpus. Further calls are made to complete a full range of services for the engine functionality (see Appendix C.1 for a complete explanation of the steps).

The semantic engine proceeds by removing unwanted corpora entries in order to resolve the valuable information entries. This occurs through many rounds of improvement. For example, the first conditioning sequence parses the ‘wordsin’ file and removes any word token that is less than 2 characters in length, providing an output that overwrites the ‘wordsin’ file. It also removes any punctuation that has been tokenized. This results in many improvement reports, such as, Input 6.1.1.8 shows 8,485 tokenized words have been removed, with 24,737 tokenized words remaining. A visual a plot of the first 20 words of the highest frequency are shown in Output 6.1.1.10.

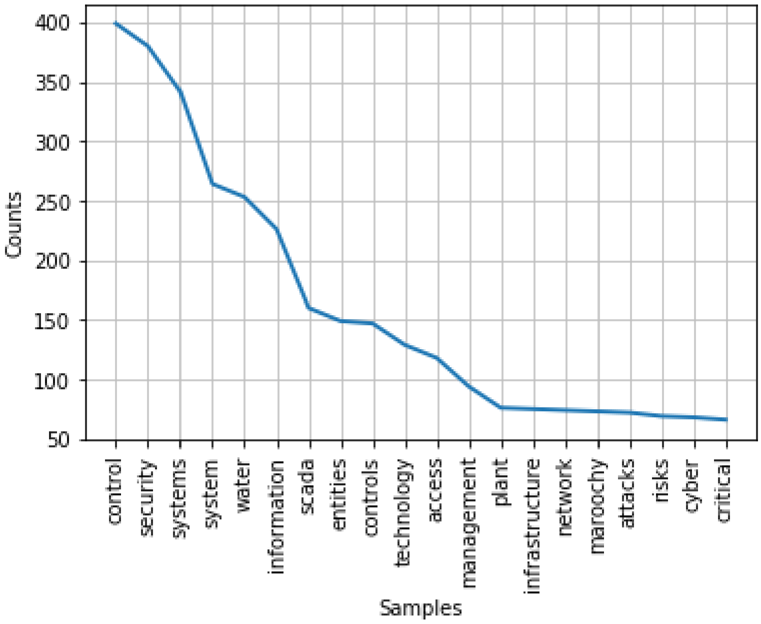
Output 6.1.1.10



Output 6.1.1.10 demonstrates the ongoing problem of worthless entries the algorithm must solve by successive improvements. It shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Maroochy corpora. The initial plot indicates that there are word tokens that do not add any meaningful input (for example;

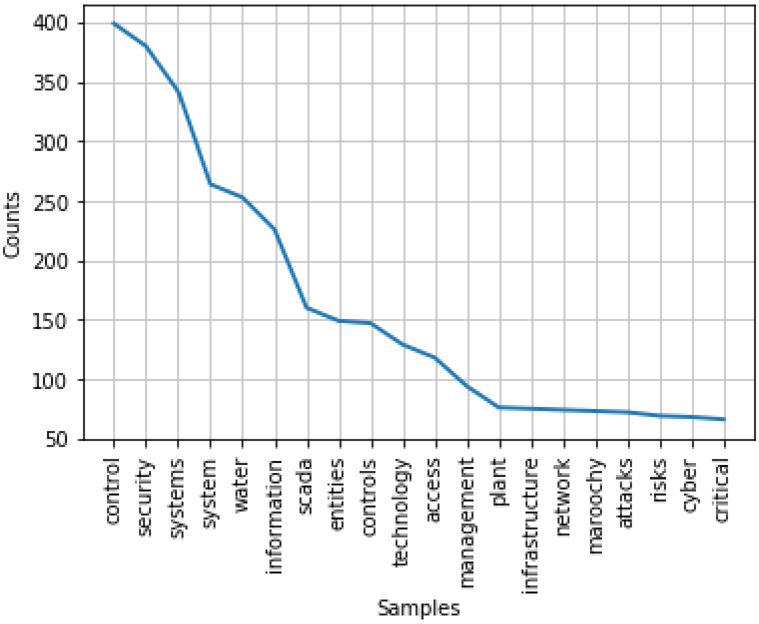
‘the’ and ‘and’). The output permits identification of words that do not add value to the experiment process. Output 6.1.1.11 indicates that there are 11 words that can be removed from the top 20 occurring words, without removing additional ‘meaning’ The identified word tokens are indicated with ‘***’. The out 6.1.1.11 list also shows that words beginning with a capital letter are distinct for words without a capital letter, as seen with the word ‘The’ and ‘the’ in the out 6.1.1.11 list (see Appendix C.1 for the specific examples used here). After improvement rounds 189 tokenized word entries have been removed from the Maroochy corpus. This is the first phase of the comprehensive conditioning sequence, and the top 20 chart now looks like:

Output 6.1.1.31



Output 6.1.1.31 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Maroochy corpora file, after stopwords have been removed, labelled stopped1. The initial plot indicates that the stop word tokens have been successfully removed, a much more refined sample obtained.

Output 6.1.1.65



Output 6.1.1.65 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Maroochy corpora file, after the conditioning process, labelled fdist4. The top 50 entries at this stage of refinement are:

Output 6.1.1.66

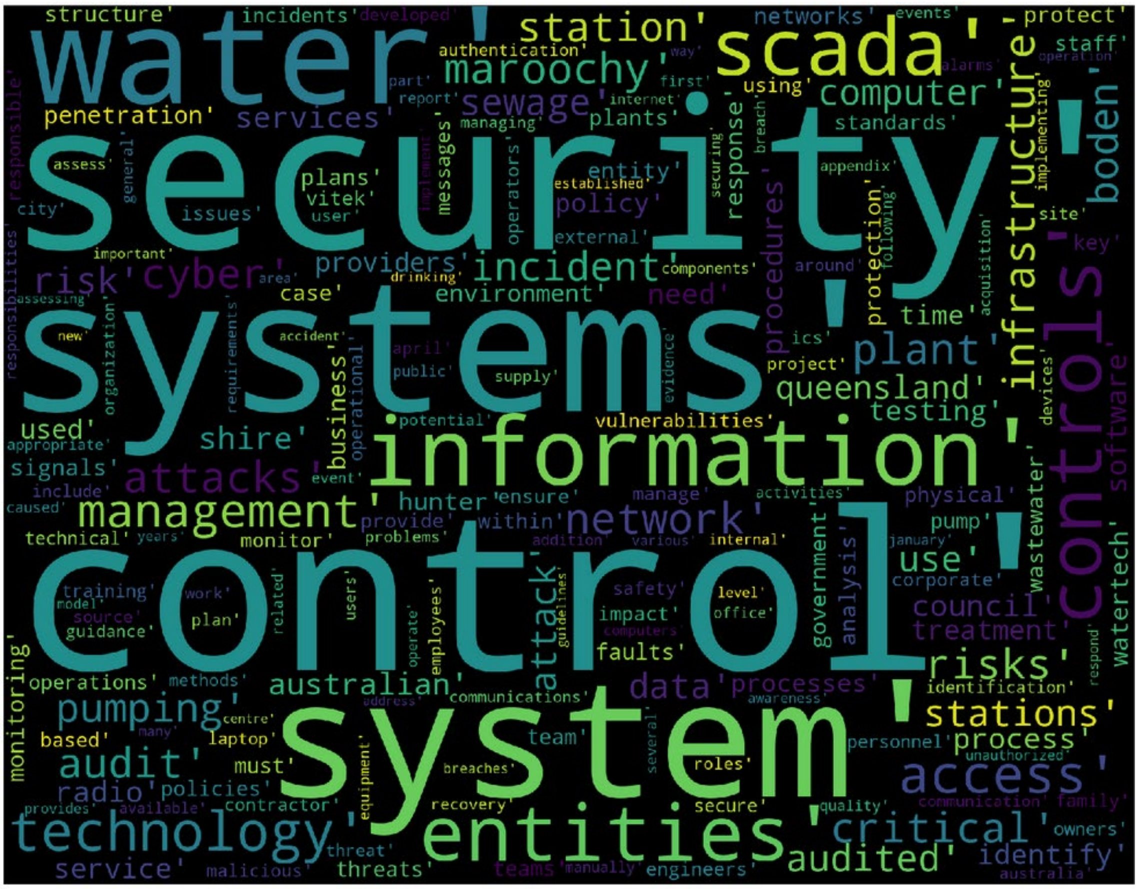
```
('control', 399),            ('audited', 54),
('security', 380),           ('sewage', 53),
('systems', 342),            ('stations', 53),
('system', 264),             ('computer', 51),
('water', 253),              ('risk', 51),
('information', 226),        ('attack', 50),
('scada', 160),              ('use', 50),
('entities', 149),           ('data', 49),
('controls', 147),           ('queensland', 48),
('technology', 129),         ('shire', 47),
('access', 118),             ('australian', 46),
('management', 94),          ('council', 45),
('plant', 76),               ('services', 45),
('infrastructure', 75),      ('procedures', 44),
('network', 74),             ('process', 44),
('Maroochy', 73),            ('radio', 44),
('attacks', 72),             ('software', 43),
('risks', 69),               ('identify', 42),
('cyber', 68),               ('need', 42),
('critical', 66),            ('service', 42),
('audit', 64),               ('testing', 42),
('station', 64),             ('treatment', 41),
('boden', 63),               ('providers', 39),
('pumping', 62),             ('used', 39),
('incident', 61),            ('policy', 38)
```

Output 6.1.1.66 shows the output of the 50 most common entries, and the corresponding word-count value of ‘stopped6’. The file ‘stopped 6’ is the final tokenized output of the conditioning sequences. In the following sub-sections other selection methods are to be applied to further refine the risk target words.

### 6.1.2 WordCloud Visualization

The next step is to implement further training and conditioning processes. The output of the training process presents conditioned data, to inform the final risk maturity enumeration. The output from Section 6.1.1 is exhibited in the form of a WordCloud visualization. Using the Word Cloud output as a visualization tool, further worthless words may be identified and removed. The training presents a conditioned output that allows further reduction of 'noise' words, as identified in the Section 6.1.1 processes. The WordCloud visualization presented as Output 6.1.2.1 shows that the most frequent words 'control', 'security' and 'systems' add value to the risk identification process.

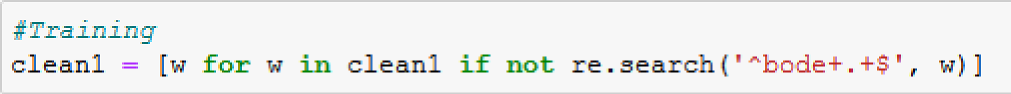
Output 6.1.2.1



### 6.1.3 Training

The output from the WordCloud visualization (Output 6.1.2.1) shows that the words 'boden', 'maroochy' 'queensland' and 'shire' are identified as 'noise' and can be removed from the word frequency output. The final output presents a trained word frequency list that is conditioned to better inform the final step of maturity modelling. The following steps show the training process.

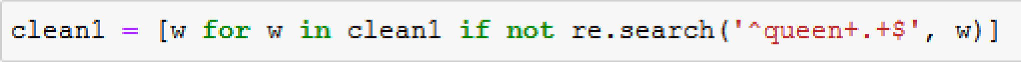
Input 6.1.3.1



```
#Training
clean1 = [w for w in clean1 if not re.search('^bode+.+$', w)]
```

Input 6.1.3.1 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘bode’, which includes the word ‘boden’. The output is saved as a tokenized file labelled ‘clean1’ overwriting the existing file.

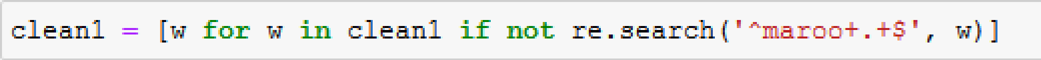
Input 6.1.3.2



```
clean1 = [w for w in clean1 if not re.search('^queen+.+$', w)]
```

Input 6.1.3.2 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘queen’, which includes the word ‘queensland’. The output is saved as a tokenized file labelled ‘clean1’ overwriting the existing file.

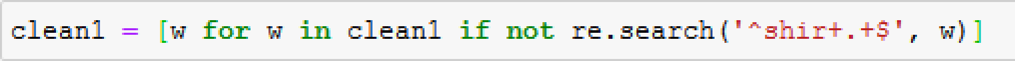
Input 6.1.3.3



```
clean1 = [w for w in clean1 if not re.search('^maroo+.+$', w)]
```

Input 6.1.3.3 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘maroo’, which includes the words ‘maroochy and ‘maroochydore’. The output is saved as a tokenized file labelled ‘clean1’ overwriting the existing file.

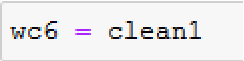
Input 6.1.3.4



```
clean1 = [w for w in clean1 if not re.search('^shir+.+$', w)]
```

Input 6.1.3.4 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘shir’, which includes the word ‘shire’. The output is saved as a tokenized file labelled ‘clean1’ overwriting the existing file.

Input 6.1.3.5

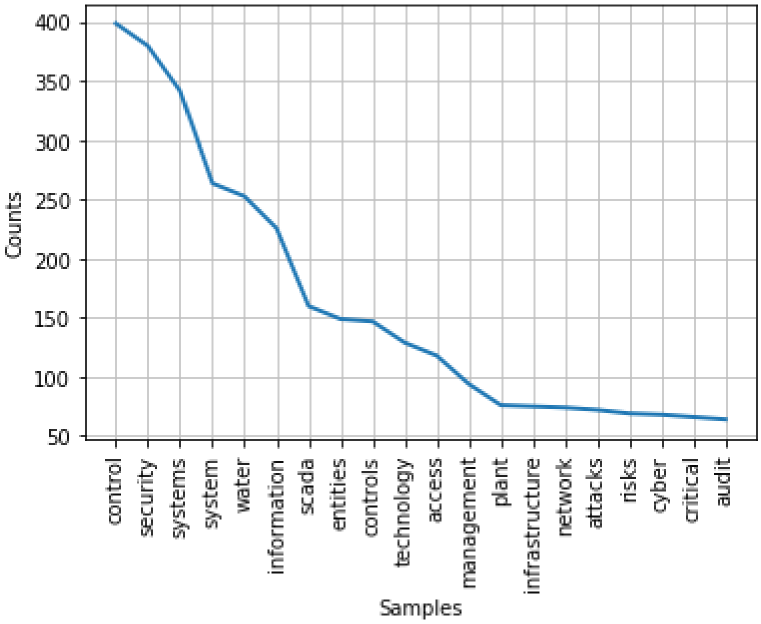


```
wc6 = clean1
```

The conditioned tokenized file ‘clean1’ is saved as ‘wc6’ by Input 6.1.3.5 for WordCloud generation.

Hence, a more refined and targeted set of risks are now identified in Output 6.1.3.7. It presents a graph of the frequency count of the first 20 words contained within the conditioned file wc6, that has experienced the visualization refinement process.

Output 6.1.3.7



Output 6.1.3.7 shows a frequency count plot of the first 20 words, when categorized by frequency, contained within the Maroochy corpora file, after training and, labelled wc6. The plot output indicates that the word token ‘maroochy’ has been successfully removed.

Input 6.1.3.10 calls for a list of the 50 most common words remaining within the Maroochy corpus conditioned and trained file, identified by frequency, providing the associated word count for each tokenized word.

Output 6.1.3.10

```
('control', 399),
('security', 380),
('systems', 342),
('system', 264),
('water', 253),
('information', 226),
('scada', 160),
('entities', 149),
('controls', 147),
('technology', 129),
('access', 118),
('management', 94),
('plant', 76),
('infrastructure', 75),
('network', 74),
('attacks', 72),
('risks', 69),
('cyber', 68),
('critical', 66),
('audit', 64),
('station', 64),
('pumping', 62),
('incident', 61),
('audited', 54),
('sewage', 53),
('stations', 53),
('computer', 51),
('risk', 51),
('attack', 50),
('use', 50),
('data', 49),
('australian', 46),
('council', 45),
('services', 45),
('procedures', 44),
('process', 44),
('radio', 44),
('software', 43),
('identify', 42),
('need', 42),
('service', 42),
('testing', 42),
('treatment', 41),
('providers', 39),
('used', 39),
('policy', 38),
('hunter', 37),
('processes', 37),
('response', 37),
('time', 37)]
```

Output 6.3.3.8 shows a list of the 50 most common entries, along with the corresponding word-count value. The output represents the final SAE output to inform the following Maturity Analysis process.

### 6.1.4 Maturity Analysis

Clustering is a process applied to the taxonomic results as defined in Chapter three. The process locates natural groups and spans by interpreting the data presented in the final refined output. The process locates areas of density where the data points are the frequency count value, and the data points have a boundary extent. The methods are described after each Table.

***Table 6.1.1: Cluster 1***

| Risk | Frequency | Maturity |
|---|---|---|
| 'control' | 399 | 5 |
| 'security' | 380 | 5 |
| 'systems' | 342 | 5 |
| 'system' | 264 | 5 |
| 'water' | 253 | 5 |
| 'information' | 226 | 5 |

Table 6.1.1 shows the data points that are contained between the boundary extent 399-226. The boundary extents are measured by identifying the largest gap between data points, which is 66, denoting the lower boundary point in Cluster 1 (226) and the upper boundary point of Cluster 2 (160). All the data points in Cluster 1 have a numeric distance of less than 66.

***Table 6.1.2: Cluster 2***

| Risk | Frequency | Maturity |
|---|---|---|
| 'scada' | 160 | 4 |
| 'entities' | 149 | 4 |
| 'controls' | 147 | 4 |

Table 6.1.2 shows the data points that are contained between the boundary extent 160-147. The boundary extents are measured by identifying the largest gap, smaller than 66 (see Table 6.1.1) between data points, which is 18, denoting the lower boundary point in Cluster 2 (147) and the upper boundary point of Cluster 3 (129). All the data points in Cluster 2 have a numeric distance of less than 18, containing greater than 2 entries.

***Table 6.1.3: Cluster 3***

| Risk | Frequency | Maturity |
|---|---|---|
| 'technology' | 129 | 3 |
| 'access' | 118 | 3 |
| 'management' | 94 | 3 |

Table 6.1.3 shows the data points that are contained between the boundary extent 129-94. The boundary extents are measured by identifying the largest gap, smaller than or equal to 18 (see Table 6.1.2) between data points, which is 18, denoting the lower boundary point in Cluster 3 (94) and the upper boundary point of Cluster 4 (76). All the data points in Cluster 3 have a numeric distance of less than 18, containing greater than 2 entries.

***Table 6.1.4: Cluster 4***

| Risk | Frequency | Maturity |
|---|---|---|
| 'plant' | 76 | 2 |
| 'infrastructure' | 75 | 2 |
| 'network' | 74 | 2 |
| 'attacks' | 72 | 2 |

Table 6.1.4 shows the data points that are contained between the boundary extent 76-72. The boundary extents are measured by identifying the largest gap, smaller than 18 (see Table 6.1.3) between data points, which is 3, denoting the lower boundary point in Cluster 4 (72) and the upper boundary point of Cluster 5 (69). All the data points in Cluster 3 have a numeric distance of less than 3, with a cluster size greater than 2.

***Table 6.1.5: Cluster 5***

| Risk | Frequency | Maturity |
|---|---|---|
| 'risks' | 69 | 1 |
| 'cyber' | 68 | 1 |
| 'critical' | 66 | 1 |
| 'audit' | 64 | 1 |
| 'station' | 64 | 1 |
| 'pumping' | 62 | 1 |
| 'incident' | 61 | 1 |

Table 6.1.5 shows the data points that are contained between the boundary extent 69-62. The boundary extents are measured by identifying the largest gap, between the remaining data points, which is 7, denoting the lower boundary point in Cluster 5. All the data points in Cluster 5 have a numeric distance of less than or equal to 2, with a cluster size greater than 2.

## 6.2 TARGET

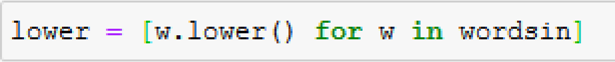
The Target Private Information Breach (Target) test case provides the basis of the second SAE automation process for the research. The target test case occurred between Nov 27th and Dec 18th 2013 resulting in the loss of 40 million credit and debit card numbers. The Information Vocabulary Text Database information includes information from forums including American legal proceedings, United States Congress proceedings and Peer-reviewed journal papers. The forums provide the information links, and the Information Vocab, in a corpus of 129,331 words. The corpora informs the SAE process throughout each process and sub-process step. The results are subject to a word-frequency analysis. A Word-Cloud visualization is then applied, to inform further training steps. The final stage of the SAE process provides a domain relevant term extraction after a word frequency cluster analysis. The final sub-section gives the maturity level computation from the Target case study and the maturity levels a business manager uses to take actions to mitigate risk. The following sub-sections are summaries of the full data sets for Target which are found in Appendix C.2.

### 6.2.1 Process

The processes applied to resolve the text copora into useful risk attributes for Target is described in step-by-step detail in Appendix C.2. The following points are a brief review of the substantial body of work provided in the Appendix C.

The overview the design for this work is given in Figure 1.1. Packages and modules are called to perform the various functions within the semantic analysis engine. For example, the **sys** module is imported to provide the Python interpreter information regarding constants, functions and methods, when using the ‘dir(system)’ call. The **codecs** module defines the base classes for standard Python encoders and decoders (codecs). The **codecs** module allows the Python interpreter to encode text to bytes and also encode text to text and bytes to bytes. The error handling lookup process is also managed by the **codecs** module. The **nltk** import command provides access to the Natural Language Tool Kit suite of text processing libraries. The libraries of nltk modules include the tokenization, word frequency and list of default stopwords. The **re** import command provides the Python interpreter regular expression matching operations. Both Unicode strings and 8-bit strings can be searched and matched. Import **os** allows the Python interpreter to interface with the Windows or Linux operating system, enabling cross platform functionality.

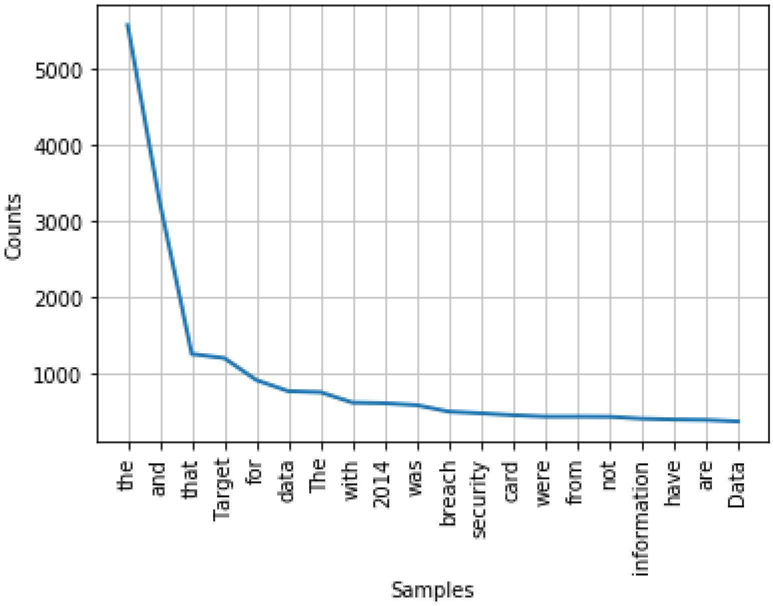
After a primary words count the conditioning sequences parse and remove insignificant words and characters in a cyclic fashion. For example, any word token that is less than 2 characters in length or any punctuation marks, and so on are successfully filtered out. (Appendix C.2) for example shows that 37,272 tokenized words have been removed, with 92,059 tokenized words remaining. Graphic outputs are also used to identify progress in the resolution processes. For example, the two plots that follow show initially many insignificant words (the, and, that, and so on), but in the second plot these are removed. All processes proceed in a cyclic fashion so that the selection of words become more and more relevant to the risk attributes. For example, Input 6.2.1.12:



```
lower = [w.lower() for w in wordsin]
```

; parses the tokenized wordlist using the call **lower()** to convert all the word tokens to lowercase. The output is saved as ‘lower’ and avoids duplication or duplicate counting of words with the same risk values.

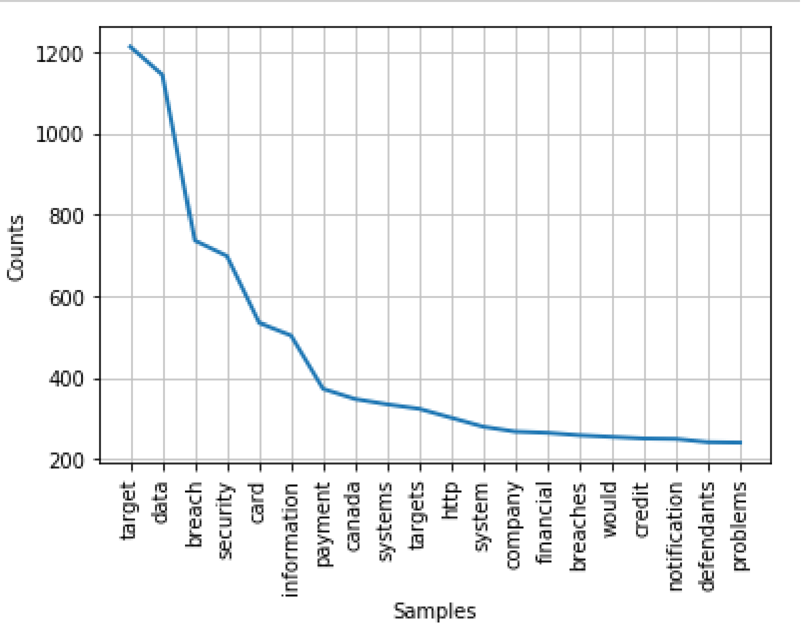
Output 6.2.1.10



Output 6.2.1.10 shows a plot of the first 20 words during resolution processes when categorized by frequency, from the Target corpora. The initial plot indicates that there are word tokens that do not add any meaningful input and do not add value to the experimental process. Output 6.2.1.11 indicates that there are 13 words that can be removed from the top 20 occurring words, without removing additional ‘meaning’ (The identified word tokens are indicated by ‘***’ in the Appendix C.2 full report). When no further words have been removed, further visual checks are made, and then the output

stabilized for maturity level ranking. The Output 6.2.1.65 chart shows a more advanced resolution of the Target data.

Output 6.2.1.65



The final resolution for Target is Output 6.2.1.66 that shows the top 50 entries contained within the ‘stopped6’ tokenized file.

Output 6.2.1.66

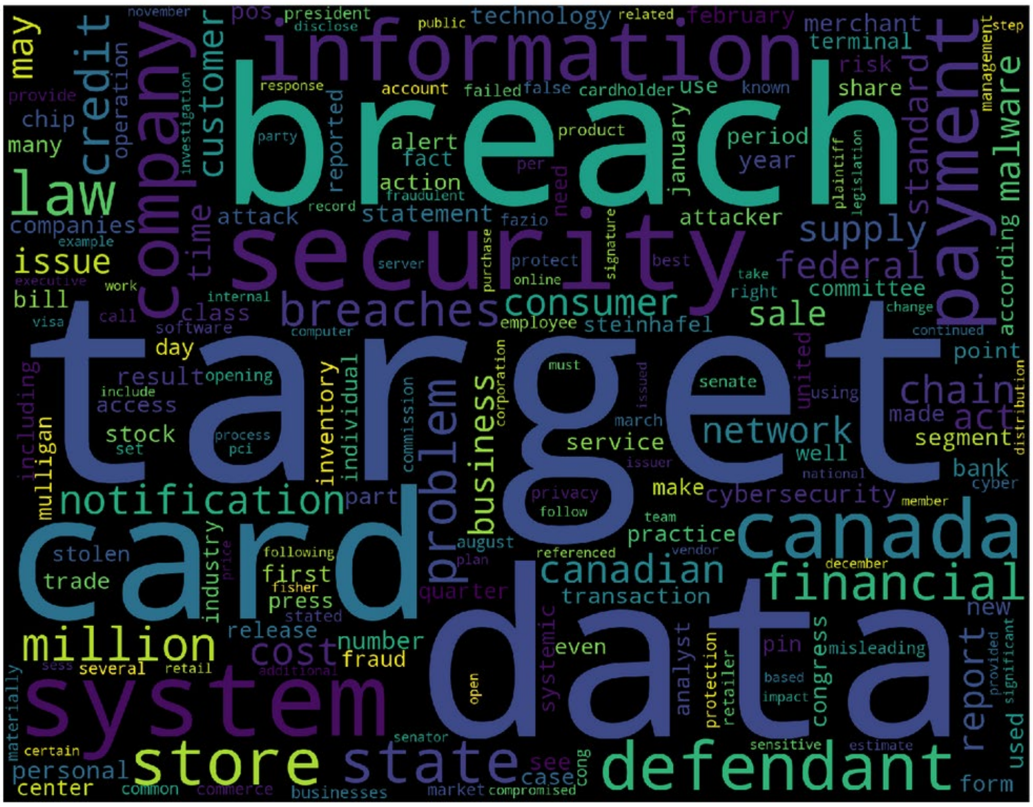
```
('target', 1212),
('data', 1143),
('breach', 736),
('security', 698),
('card', 535),
('information', 504),
('payment', 373),
('canada', 348),
('systems', 335),
('targets', 324),
('system', 280),
('company', 268),
('financial', 265),
('breaches', 259),
('credit', 251),
('notification', 250),
('defendants', 242),
('problems', 241),
('canadian', 238),
('million', 238),
('chain', 236),
('stores', 231),
('cards', 210),
('supply', 208),
('federal', 206),
('act', 185),
('malware', 185),
('business', 175),
('law', 168),
('cybersecurity', 164),
('network', 164),
('personal', 157),
('report', 149),
('companies', 145),
('customers', 145),
('states', 145),
('congress', 144),
('new', 142),
('used', 141),
('technology', 138),
('class', 135),
('state', 135),
('costs', 134),
('made', 133),
('first', 131),
('canadas', 130),
('inventory', 127),
('see', 127),
('pos', 125),
('issues', 121)]
```

Output 6.2.1.66 shows the output of the 50 most common entries, and the corresponding word-count value of ‘stopped6’. The file ‘stopped 6’ is the final tokenized output of the stopword conditioning sequences for the Target data.

### 6.2.2 WordCloud Visualisation

The next step is to implement further training and conditioning processing. The output of the training process presents conditioned data, to inform the final risk maturity enumeration. The output from Section 6.2.1 is exhibited in the form of a WordCloud visualization. Using the Word Cloud output as a visualization tool, Output 6.2.2.1 shows that the word ‘target has been identified as the word that occurs most frequently. As the word target does not add benefit to the risk identification process, the word visualization signals further reductions. The training presents a conditioned output that allows further reduction of ‘noise’ words, as identified in the Section 6.2.1. The following image (output 6.2.2.1) is an example of data visualization that informs the resolution processes. These further perspectives on the data allow supervision or automation to occur but lead to greater efficiencies in resolving the data towards the key risks. Visualization also allows checking and self-auditing as the processes proceed. The visualization perspective shows a different angle on the same data and can pick up wastage that is missed in the text brokerage parts of the algorithms.

Output 6.2.2.1

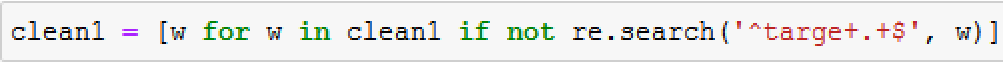


The WordCloud visualization presented as Output 6.3.2.1 shows that the word ‘target’ is identified as the word that occurs most frequently.

### 6.2.3 Training

Using the Word Cloud output as a visualization tool, Output 6.2.2.1 shows that the word ‘target’ has been identified as the word that occurs most frequently. As the word ‘target’ does not add benefit to the risk identification process, the word is a good example to demonstrate the training abilities of the prototype. The word ‘breach’ is also selected to demonstrate the conditioning process, as the word does not add benefit to the risk identification process (the event is loosely described as a ‘breach’ in all the publications). The training presents a conditioned output that allows further reduction of ‘noise’ words, as identified in the Section 6.2.1 process.

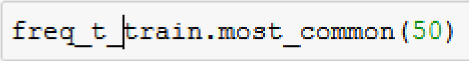
Input 6.2.3.1



```
clean1 = [w for w in clean1 if not re.search('^targe+.+$', w)]
```

Input 6.2.3.1 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘target’, which includes the words ‘target’ and ‘targets’. The output is saved as a tokenized file labeled ‘clean1’, overwriting the existing file and presents an advanced resolution from the earlier output.6.2.1.66.

Input 6.2.3.10



```
freq_t_train.most_common(50)
```

Input 6.2.3.10 calls for a list of the 50 most common words remaining within the Target corpus conditioned and trained file, identified by frequency, providing the associated word count for each tokenized word.

Output 6.2.3.10

```
('data', 1143),
('security', 698),
('card', 535),
('information', 504),
('payment', 373),
('systems', 335),
('system', 280),
('company', 268),
('financial', 265),
('credit', 251),
('notification', 250),
('defendants', 242),
('problems', 241),
('million', 238),
('chain', 236),
('stores', 231),
('cards', 210),
('supply', 208),
('federal', 206),
('act', 185),
('malware', 185),
('business', 175),
('personal', 157),
('report', 149),
('companies', 145),
('customers', 145),
('states', 145),
('congress', 144),
('new', 142),
('used', 141),
('technology', 138),
('class', 135),
('state', 135),
('costs', 134),
('made', 133),
('first', 131),
('inventory', 127),
('see', 127),
('pos', 125),
('issues', 121),
('segment', 120),
('laws', 119),
('consumers', 117),
('including', 117),
('law', 168),                   ('systemic', 117),
('cybersecurity', 164),         ('february', 116),
('network', 164),               ('access', 115)]
```

Output 6.2.3.10 shows the output of the 50 most common entries in the Target corpus after conditioning and training, along with the corresponding word-count value. The output represents the final SAE output is to inform the following Maturity Analysis process.

### 6.2.4 Maturity Analysis

Clustering is a process applied to the taxonomic results as defined in Chapter three. The process locates natural groups by interpreting the data presented with Output 6.2.3.10. The process locates areas of density where the data points are the frequency count value, and the data points have a boundary extent. The following Tables take these outputs and compute the maturity values as described by cluster and span analysis in Chapter three.

***Table 6.2.1: Cluster 1***

| Risk | Frequency | Maturity |
|---|---|---|
| 'data' | 1143 | 5 |
| 'security' | 698 | 5 |
| 'card' | 535 | 5 |
| 'information' | 504 | 5 |
| 'payment' | 373 | 5 |
| 'systems' | 335 | 5 |

Table 6.3.1 shows the data points that are contained between the boundary extent 1143-335. The boundary extents are measured by identifying the largest gap between data points, which is 55, denoting the lower boundary point in Cluster 1 and the upper boundary point of Cluster 2. All the data points in Cluster 1 have a numeric distance of less than 55, and a cluster of more than 2 items. Thus, the first two entries are included in Cluster 1.

***Table 6.2.2: Cluster 2***

| Risk | Frequency | Maturity |
|---|---|---|
| 'system' | 280 | 4 |
| 'company' | 268 | 4 |
| 'financial' | 265 | 4 |
| 'credit' | 251 | 4 |
| 'notification' | 250 | 4 |

Table 6.2.2 shows the data points that are contained between the boundary extent 280-250. The boundary extents are measured by identifying the largest gap, smaller than 55 (see Table 6.3.1) between data points, which is 8, denoting the lower boundary point in Cluster 2 (250) and the upper boundary point of Cluster 3 (242). All the data points in Cluster 2 have a numeric distance of less than 8, containing a cluster of more than 2 items.

***Table 6.2.3: Cluster 3***

| Risk | Frequency | Maturity |
|---|---|---|
| 'defendants' | 242 | 3 |
| 'problems' | 241 | 3 |
| 'million' | 238 | 3 |
| 'chain' | 236 | 3 |
| 'stores' | 231 | 3 |

Table 6.2.3 shows the data points that are contained between the boundary extent 242-231. The boundary extents are measured by identifying the largest gap between data points, which is 21, denoting the lower boundary point in Cluster 3 (231) and the upper boundary point of Cluster 4 (210). All the data points in Cluster 3 have a numeric distance of less than 21.

***Table 6.2.4: Cluster 4***

| Risk | Frequency | Maturity |
|---|---|---|
| 'cards' | 210 | 2 |
| 'supply' | 208 | 2 |
| 'federal' | 206 | 2 |

Table 6.2.4 shows the data points that are contained between the boundary extent 210-208. The boundary extents are measured by identifying the largest gap between the remaining data points, which is 21, denoting the lower boundary point in Cluster 4 (206) and the upper boundary point of Cluster 2 (185). All the data points in Cluster 3 have a numeric distance of less than 21, with a cluster size greater than 2.

*Table 6.2.5: Cluster 5*

| Risk | Frequency | Maturity |
|---|---|---|
| 'act' | 185 | 1 |
| 'malware' | 175 | 1 |
| 'business' | 168 | 1 |
| 'law' | 164 | 1 |
| 'cybersecurity' | 164 | 1 |
| 'network' | 164 | 1 |
| 'personal' | 157 | 1 |

Table 6.2.5 shows the data points that are contained between the boundary extent 185-157. The boundary extents are measured by identifying the largest gap between the remaining data points, which is 8, denoting the lower boundary point in Cluster 5 (157) All the data points in Cluster 5 have a numeric distance of less than 8, with a cluster size greater than 2

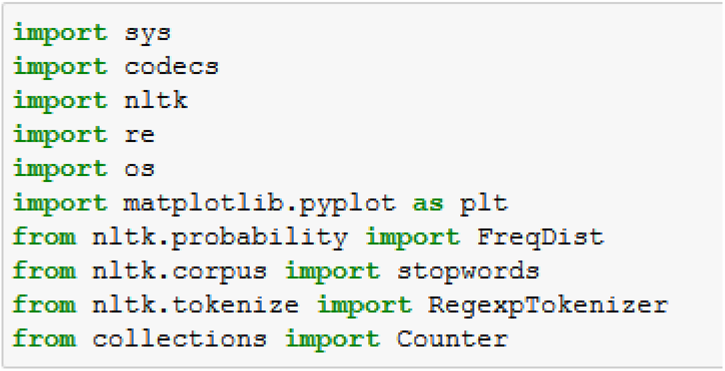
## 6.3 TESLA

The CANbus and Vehicle Automation (tesla) test case informs the third SAE automated process of the research. The tesla test case is the most recent of the three test cases, involving an accident which occurred occurring in May 2016. The Information Vocabulary Text Database information included information from forums including American National Transportation Board reports; IEEE Peer-reviewed journal papers; and Industry Best Practice reports. The forums provide the information links, supplying the Information Vocab, introduced as a corpus of 67,272 words. The corpora informs the SAE process throughout each process and sub-process step. The results are subject to a word-frequency analysis. A Word-Cloud visualization is then applied, to inform further training steps. The final stage of the SAE process provides a domain relevant term extraction after a cluster analysis. These analyses deliver the maturity values a business manager requires for decision-making (see 6.3.4). The following sub-sections are summaries of the full data sets for Tesla found in Appendix C.3.

### 6.3.1 Process

The processes of the Natural Language algorithms are captured in the semantic engine construct. This sub-section provides a brief overview of the Input and Output loops that resolve and rank the relevant words.

Input 6.3.1.1



```
import sys
import codecs
import nltk
import re
import os
import matplotlib.pyplot as plt
from nltk.probability import FreqDist
from nltk.corpus import stopwords
from nltk.tokenize import RegexpTokenizer
from collections import Counter
```

Input 6.3.1.1 shows the packages and modules that will be required to perform the various functions within the semantic analysis engine. The **sys** module is imported to provide the Python interpreter information regarding constants, functions and methods, when using the ‘dir(system)’ call. The **codecs** module defines the base classes for standard Python encoders and decoders (codecs). The **codecs** module allow the Python interpreter to encode text to bytes and also encode text to text and bytes to bytes. The error handling lookup process is also managed by the **codecs** module. The **nltk** import command provides access to the Natural Language Tool Kit suite of text processing libraries. The libraries of nltk modules include the tokenization, word frequency and list of default stopwords. The **re** import command provides the Python interpreter regular expression matching operations. Both Unicode strings and 8-bit strings can be searched and matched. Import **os** allows the Python interpreter to interface with the Windows or Linux operating system, enabling cross platform functionality.

The purpose of an Input is to confirm the word-breaking process has been implemented correctly, by retrieving more and more relevant words as the processes persist, for example:

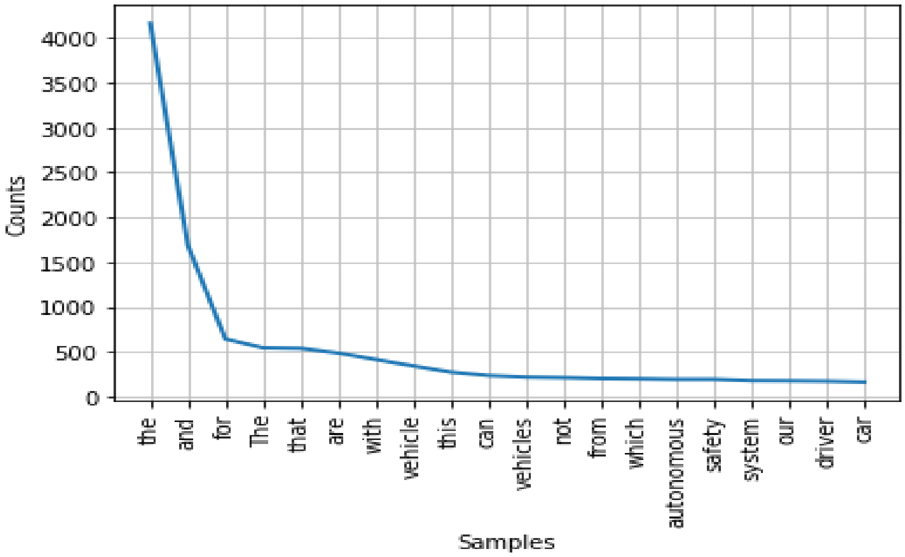
Output 6.3.1.5

```
'Assessment',
'and',
'standardization',
'of',
'autonomous',
'vehicles',
'Abstract',
'Autonomous',
'vehicle',
'technology',
'presents',
'a',
'huge',
'challenge',
'to',
'standardization',
'and',
'legal',
'bodies',
```

Output 6.3.1.5 shows the first 20 words of the Tesla corpus, correctly identifying each separate word, therefore showing that the word-breaking process has occurred. These and other more complex resolution processes occur until a primary word count of the entire corpus is completed with no conditioning or training. The word count output allocates a baseline figure that is used to determine the effectiveness of each subsequent training of a conditioning sequence. For example, it can remove any word token that is less than two characters in length, and any punctuation that has been tokenized. A further deliverable is charted visualization that reports the top twenty filtered words at each iteration of the algorithm. Breaking into the algorithm work cycle at any point shows, for example that 20,303 tokenized words have been removed, with 46,969 tokenized words remaining, when compared to the previous output. A plot chart of the first 20 words within the current level of resolution are displayed as Output 6.3.1.10.

Output 6.3.1.10 shows a plot of the first 20 words, when categorized by frequency, contained within the Tesla corpora. The initial plot indicates that there are word tokens that do not add any meaningful input, and hence the action taken is to invoke further iterations of the automated algorithm. The final iteration shows the output of the 50 most common entries, and the corresponding word-count value. The file 'stopped x' is the final tokenized output of the stopword conditioning sequences. For a full explanation and demonstration of the processes used in the Tesla case study refer to Appendix C.3.1 for all data.

Output 6.3.1.10

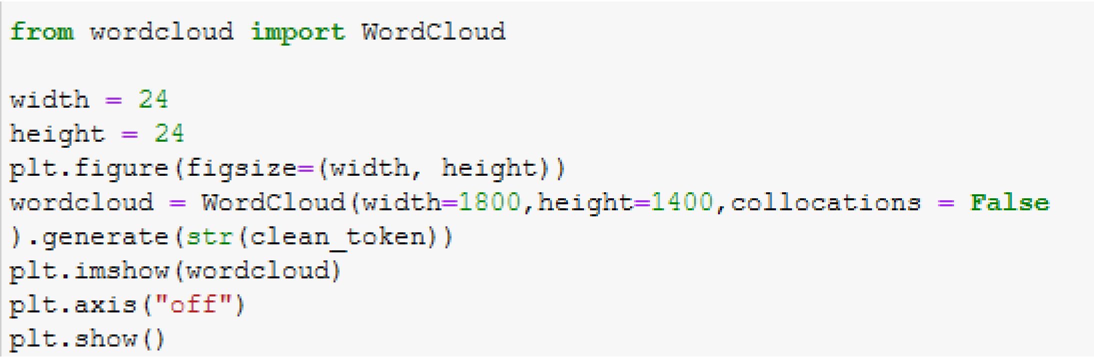


### 6.3.2 WordCloud Visualisation

The next step is to implement further training and conditioning processing. The output of the training process presents conditioned data, to inform the final risk maturity enumeration. The output from Section 6.3.1 is exhibited in the form of a WordCloud visualization. Using the Word Cloud output as a visualization tool, Output 6.3.2.1 shows that the word ‘vehicle’ has been identified as the word that occurs most frequently. As the word ‘vehicle does not add benefit to the risk identification process, the word is a good example to demonstrate the training abilities of the prototype. The training presents a conditioned output that allows further reduction of ‘noise’ words, as identified in the Section 6.3.1 processes.

The following example illustrates the input and visual output sequence that follows from iterative stages in the algorithm.

Input 6.3.2.1

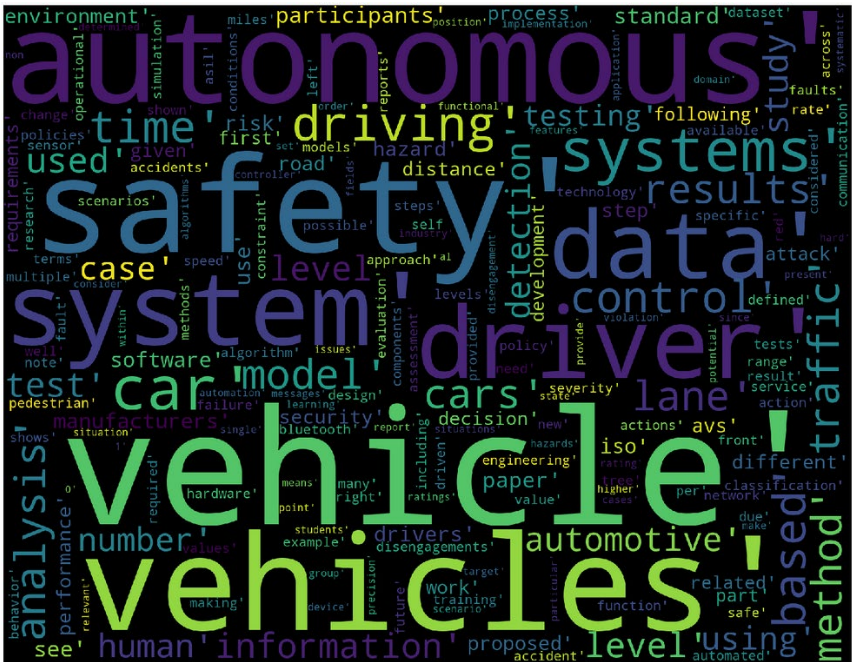


```
from wordcloud import WordCloud

width = 24
height = 24
plt.figure(figsize=(width, height))
wordcloud = WordCloud(width=1800,height=1400,collocations = False
).generate(str(clean_token))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Input 6.3.2.1 calls for the WordCloud module and defines the arguments to be implemented. The arguments define parameters of size, position and figure size.

Output 6.3.2.1



The WordCloud visualization presented as Output 6.3.2.1 shows that the words ‘vehicle’ is identified as the word that occurs most frequently. The visualization also shows that the word ‘vehicles’ also occurs frequently. These words are not helpful and can now be filtered.

### 6.3.3 Training

The output from the WordCloud visualization (Output 6.3.2.1) shows that the words ‘vehicle’ and ‘vehicles’ are identified as ‘noise’ and can be removed from the word frequency output. This is done by automated processes after identification. The final output presents a trained word frequency list that is conditioned to better inform the final step of maturity modelling. Input 6.3.3.8 calls for a list of the 50 most common words remaining within the Tesla corpus conditioned and trained file and identified by word count for each tokenized word. For a full explanation and demonstration of the processes used in the Tesla case study refer to Appendix C.3 for all the data and training steps.

Output 6.3.3.8

```
('safety', 236),             ('using', 97),
('autonomous', 222),         ('number', 96),
('system', 201),             ('level', 95),
('driver', 184),             ('testing', 93),
('data', 168),               ('case', 91),
('systems', 168),            ('participants', 87),
('car', 167),                ('see', 85),
('control', 135),            ('standard', 82),
('driving', 134),            ('use', 82),
('time', 128),               ('paper', 81),
('traffic', 123),            ('manufacturers', 79),
('analysis', 122),           ('drivers', 78),
('method', 121),             ('iso', 76),
('based', 115),              ('risk', 76),
('model', 115),              ('software', 76),
('lane', 114),               ('environment', 74),
('cars', 113),               ('hazard', 74),
('results', 111),            ('proposed', 71),
('information', 108),        ('step', 69),
('used', 106),               ('process', 68),
('detection', 103),          ('distance', 67),
('study', 103),              ('road', 67),
('automotive', 102),         ('different', 66),
('test', 102),               ('security', 66),
('human', 99),               ('performance', 65
```

Output 6.3.3.8 shows the output of the 50 most common entries in the Tesla corpus after conditioning and training, and the corresponding word-count value. The output represents the final SAE output to inform the following Maturity Analysis process.

### 6.3.4 Maturity Analysis

Clustering is a process applied to the taxonomic results is defined in Chapter three. The process locates natural groups and spans by interpreting the data presented in Output 6.3.3.8. The process locates areas of high density and the boundary points between areas. The clustering phenomena is the result of the algorithmic resolution processes and groups the data into naturally segregated rankings. These rankings are then translated into maturity levels that are ranked on a scale of one to five (five is the most mature). The following Tables review, analyze, and rank the data of the final output 6.3.3.8. Each Table show the associated risks and the system readiness for mitigation.

***Table 6.3.1: Cluster 1***

| Risk | Frequency | Maturity |
|---|---|---|
| 'safety' | 236 | 5 |
| 'autonomous' | 222 | 5 |
| 'system' | 201 | 5 |
| 'driver' | 184 | 5 |
| 'data' | 168 | 5 |
| 'systems' | 168 | 5 |
| 'car' | 167 | 5 |

Table 6.3.1 shows the data points that are contained between the boundary extent 236-167. The boundary extents are measured by identifying the largest gap between data points, which is 32, denoting the lower boundary point in Cluster 1 and the upper boundary point of Cluster 2. All the data points in Cluster 1 have a numeric distance of less than 32.

***Table 6.3.2: Cluster 2***

| Risk | Frequency | Maturity |
|---|---|---|
| 'control' | 135 | 4 |
| 'driving' | 134 | 4 |
| 'time' | 128 | 4 |
| 'traffic' | 123 | 4 |
| 'analysis' | 122 | 4 |
| 'method' | 121 | 4 |

Table 6.3.2 shows the data points that are contained between the boundary extent 135-121. The boundary extents are measured by identifying the largest gap, smaller than 32 (see Table 6.1) between data points, which is 6, denoting the lower boundary point in

Cluster 2 and the upper boundary point of Cluster 3. All the data points in Cluster 2 have a numeric distance of less than 6.

***Table 6.3.3: Cluster 3***

| Risk | Frequency | Maturity |
|---|---|---|
| 'based' | 115 | 3 |
| 'model' | 115 | 3 |
| 'lane' | 114 | 3 |
| 'cars' | 113 | 3 |
| 'results' | 111 | 3 |
| 'information' | 108 | 3 |
| 'used' | 106 | 3 |
| 'detection' | 103 | 3 |
| test | 102 | 3 |
| 'human' | 99 | 3 |
| 'using' | 97 | 3 |
| 'number' | 96 | 3 |
| 'level' | 95 | 3 |
| 'testing' | 93 | 3 |
| 'case' | 91 | 3 |

Table 6.3.3 shows the data points that are contained between the boundary extent 115-91. The boundary extents are measured by identifying the largest gap, smaller than 6 (see Table 6.2) between data points, which is 4, denoting the lower boundary point in Cluster 3 and the upper boundary point of Cluster 4. All the data points in Cluster 3 have a numeric distance of less than 4.

***Table 6.3.4: Cluster 4***

| Risk | Frequency | Maturity |
|---|---|---|
| 'participants' | 87 | 2 |
| 'see' | 85 | 2 |
| 'standard' | 82 | 2 |
| 'use' | 82 | 2 |
| 'paper' | 81 | 2 |

Table 6.3.4 shows the data points that are contained between the boundary extent 87-81. The boundary extents are measured by identifying the largest gap, smaller than 4 (see Table 6.3) between data points, which is 3, denoting the lower boundary point in Cluster

4 and the upper boundary point of Cluster 2. All the data points in Cluster 3 have a numeric distance of less than 3, with a cluster size greater than 2.

***Table 6.3.5: Cluster 5***

| Risk | Frequency | Maturity |
|---|---|---|
| 'manufacturers' | 79 | 1 |
| 'drivers' | 78 | 1 |
| 'iso' | 76 | 1 |
| 'software' | 76 | 1 |
| 'environment' | 74 | 1 |
| 'hazard' | 74 | 1 |
| 'proposed' | 71 | 1 |
| 'step' | 69 | 1 |
| 'process' | 68 | 1 |
| 'distance' | 67 | 1 |
| 'road' | 67 | 1 |
| 'different' | 66 | 1 |
| 'security' | 66 | 1 |
| 'performance' | 65 | 1 |

Table 6.3.5 shows the data points that are contained between the boundary extent 79-65. The boundary extents are measured by identifying the largest gap, smaller than 3 (see Table 6.3) between data points, which is 2, denoting the lower boundary point in Cluster 5 All the data points in Cluster 5 have a numeric distance of less than 2, with a cluster size greater than 2.

## 6.4 CONCLUSION

Chapter six demonstrates each step of the automated Semantic Analysis Engine (SAE) process. The three test cases subjected to manual analysis in Chapter four have been processed automatically by the SAE. The Information Vocabulary, specific to each test case, has been introduced to each test case analysis iteration of the SAE. The Information Vocabulary includes documents from legal, industrial, best practice and peer reviewed sources. Each test case has been selected to be from widely divergent IoT domains, over a wide time-frame. Each test case presents a unique problem involving exploitation of risk vectors.

The process involved several iterations of entity removal, designed to filter word-entities that do not add value. The output from each test case SAE analysis is provided in terms of frequency clusters, producing and populating a domain specific Taxonomy for each test case. The resultant Taxonomy informs the maturity model, in terms of risk identification. Each of the risk vector demonstrates a risk vulnerability that will be analyzed for a maturity level, using the Oxford model (GCSCC 2014). The maturity level and identified risk vectors will be discussed in Chapter seven. Chapter seven will analyze and discuss the findings presented in Chapter six. The resultant determination from the discussion of results in Chapter seven will provide the final stage of maturity modelling and capability.

# Chapter 7
# DISCUSSION

## 7.0 INTRODUCTION

Chapter seven is designed to answer the research questions, test hypotheses, and then to discuss these findings in terms of what the thesis set out to achieve. The literature review in Chapter two provided a theoretical foundation for the research context and Chapter three the specification for research. The aim of the research is to develop a solution to the problem area identified in Chapter two, that a gap exists in cybersecurity risk maturity enumeration within the Internet of Things (IoT) domain. The inference formed during these early stages of the research, that an assessment of manual risk maturity evaluation methods may present a practical solution pathway, enabled the researcher to define the objectives of the solution artefacts. The pilot study undertaken in Chapter four evaluated the application of manual taxonomic and risk enumeration process steps to three widely divergent case studies. The evaluation informed the research that the integration of information from IoT based incidents forms a valuable information accumulation. An analysis of the process steps applied to the case studies informed the development of the artefacts that are delivered and discussed in Chapter seven. The deliverables discussed are presented as artefacts, and as output of the overarching Design Science methodology. There are two further outputs discussed as artefacts, as they provide novel solutions to the problem area. These are the initial inference, and the use of theory from the Information Systems (IS) data science domain, as an exaptation. The resultant deliverable is a Proof-of-Concept in the form of a software based, algorithmic prototype. The automated findings presented in Chapter six (and Appendix C), are discussed and used in this chapter. The Hypotheses are assessed positively with evidence from the prototype application output findings, the research sub-questions are answered, and an effective solution to the main research question is presented and discussed.

Chapter seven is structured to present and discuss the hypotheses, the research sub-questions and the main research questions in Section 7.1. Section 7.2 and Section 7.3 have a summary of key findings. The validity and reliability of the research findings is discussed in Section 7.4. The final sections discuss the implications arising from this research. Section 7.5, expresses and addresses the research limitations.

## 7.1 RESEARCH QUESTIONS TO BE ANSWERED

The following section takes each of the hypothesis from Chapter three and tests them by referencing data from Chapter six (and also Appendix C references). The tests proceed by tabulating the relevant evidence for and against the hypothesis, and then weighting the evidence to evaluate an outcome.

### 7.1.1 H1: Hypothesis One

The risk aspects can be identified, using cyber forensic and data analysis techniques, referenced from data in Chapter six (and Appendices). The information gained from investigating H1 will, in turn, provide answers to the Research Questions.

| **H1**: Risk aspects are identified, using cyber forensic and data analysis techniques. | |
|---|---|
| **Evidence For** | **Evidence Against** |
| Output 6.1.1.2, Output 6.2.1.2, and Output 6.3.1.2: Domain specific Corpora identification. Demonstrates identification of topic of focus for each domain provides the first step information accumulation. Each primary information accumulation is specific for each test case scenario. | |
| Input 6.1.1.4, Input 6.2.1.4, and Input 6.3.1.4: Initial process of the data to output a clean and standardized format ready for analysis. | |
| Output 6.1.1.6, Output 6.2.1.6, and Output 6.3.1.6: Presents an initial word count of each corpus. Demonstrates the different word count for each of the three test areas, showing the difference in output from the information links | |
| Output 6.1.1.8, Output 6.2.1.8, and Output 6.3.1.8: demonstrates the effectiveness of the initial word conditioning process for each of the three test case corpora. | |

| | |
|---|---|
| | Output 6.1.1.11, Output 6.2.1.11, and Output 6.3.1.11: Presents an initial word count frequency output. Demonstrates the initial output is not effective for semantic analysis |
| | Output 6.1.1.22, Output 6.2.1.22, and Output 6.3.1.22 Presents a word frequency output after comprehensive conditioning. Demonstrates that the comprehensive conditioning process does not provide an output suitable for semantic analysis |
| Output 6.1.1.6, Output 6.2.1.6, and Output 6.3.1.6: Presents a word count of each corpus after comprehensive conditioning and stopword removal. Demonstrates the different word count for each of the three test areas, identifying the difference in output from the conditioning processes. | |
| Output 6.1.1.32, Output 6.2.1.32, and Output 6.3.1.32: Presents a frequency word count after the conditioning and initial stopword removal process. Demonstrates the basis to provide an effectively conditioned output, suitable for initial semantic analysis. | |
| Output 6.1.1.63, Output 6.2.1.63, and Output 6.3.1.63: Presents a list of the first 20 entries within each conditioned word list, after comprehensive stopword removal iterations. Demonstrates the effectiveness of further word conditioning through stopword removal. | |
| Output 6.1.1.66, Output 6.2.1.66, and Output 6.3.1.66, Presents a frequency word count after the comprehensive | |

| | |
|---|---|
| conditioning and a final stopword removal process. Demonstrates an effectively conditioned output, suitable for semantic analysis | |

| |
|---|
| **Answer**:<br>The findings gathered from the experiment shown in Chapter six, provide evidence of identifiable risk aspects. The risk aspects are identified after application of semantic analysis techniques of cyber forensic output in the form of domain specific corpora. The two aspects that demonstrate evidence against the Hypothesis are initial process output that was in turn used to refine the semantic analysis process.<br>Thus, the conclusion is that the hypothesis cannot be rejected from the evidence in these tests. Risk aspects can be identified through semantic data techniques, to inform levels of Risk Capability Maturity for the internet of Things. |

### 7.1.2 H2: Hypothesis Two

| |
|---|
| **H2**:<br>The output from Hypothesis One testing informs a Capability Maturity Tool to identify IoT risk. |

| **Evidence For** | **Evidence against** |
|---|---|
| Output 6.1.1.8, Output 6.2.1.8, and Output 6.3.1.8: demonstrates the effectiveness of the initial word conditioning process for each of the three test case corpora. | |
| Output 6.1.1.6, Output 6.2.1.6, and Output 6.3.1.6: Presents a word count of each corpus after comprehensive conditioning and stopword removal. It demonstrates the different word count for each of the three test areas, identifying the difference in output from the conditioning process. | |
| Output 6.1.1.32, Output 6.2.1.32, and Output 6.3.1.32: Presents a frequency word count after the conditioning and initial stopword removal process. It | |

| | |
|---|---|
| demonstrates the basis to provide an effectively conditioned output, suitable for initial semantic analysis. | |
| Output 6.1.1.63, Output 6.2.1.63, and Output 6.3.1.63: Presents a list of the first 20 entries within each conditioned word list, after comprehensive stopword removal iterations. It demonstrates the effectiveness of further word conditioning through stopword removal. | |
| Output 6.1.1.66, Output 6.2.1.66, and Output 6.3.1.66, presents a frequency word count after the comprehensive conditioning and final stopword removal process. It demonstrates an effectively conditioned output, suitable for semantic analysis. | |
| Output 6.1.2.1, Output 6.2.2.1, and Output 6.3.2.1 present a WordCloud visualization of the word frequency. | |
| Output 6.1.3.22, Output 6.2.3.22, and Output 6.3.3.22 present a trained word frequency identifying risk attributes. | |
| Output 6.1.4.1, 6.1.4.2, 6.1.4.3, 6.1.4.4, and 6.1.4.5 present risk factors that inform a Maturity Model for the Maroochy case study. | |
| Output 6.2.4.1, 6.2.4.2, 6.2.4.3, 6.2.4.4, and 6.2.4.5 present risk factors that inform a Maturity Model for the Target case study. | |
| Output 6.3.4.1, 6.3.4.2, 6.3.4.3, 6.3.4.4, and 6.3.4.5 present risk factors that inform a Maturity Model for the Tesla case study | |

**Answer**:

The findings from the experiment presented in Chapter six, provide sufficient evidence that Hypothesis Two cannot be rejected. The findings provide evidence that the risk aspects identified from Hypothesis One inform the Capability Maturity Tool. Thus, the risk aspects identified through semantic data techniques as an output of Hypothesis One informs the risk capability maturity for the internet of Things.

### 7.1.3 RSQ1: Research Sub-Question One

The following sub-question data locates risk aspects in the evidence of Chapter six from the cyber forensics and semantic data analysis.

**RSQ1**:

What risk aspects are identified using cyber forensics and semantic data analysis techniques?

| **Cyber Forensic Aspects** | **Semantic Data Analysis Aspects** |
|---|---|
| Section 6.1, Sections 6.2. and Section 6.3: Selection of the information links from cyber forensic analysis deliver an information vocab corpus. | |
| Input 6.1.1.3, Input 6.2.1.3, and Input 6.3.1.3: Accesses the cyber forensic analysis corpora. | |
| Output 6.1.1.10, Output 6.2.1.10, and Output 6.3.1.10: Initial Cyber Forensic Corpora frequency analysis output. | |
| | Output 6.1.1.11, Output 6.2.1.11, and Output 6.3.11 present a broad range of risk aspects as an initial semantic data analysis technique. |
| | Output 6.1.1.22, Output 6.2.1.22, and Output 6.3.1.22 present a range of risk aspects from a conditioning round of the semantic data analysis. |

| | |
|---|---|
| | Output 6.1.1.66, output 6.2.1.66, and Output 6.3.1.66 present a range of risk aspects from a round of semantic data analysis using stopword removal techniques. |
| | Output 6.1.2.1, Output 6.2.2.1, and Output 6.3.2.1 present a Word Cloud visualization of risk aspects identified from a round of semantic data analysis using stopword removal techniques. |
| | Output 6.1.3.11, Output 6.2.3.11, and Output 6.3.3.11 present a Word Cloud visualization of risk aspects identified from completed rounds of semantic data analysis using training techniques. |
| | Output 6.1.3.22, Output 6.2.3.22, and Output 6.3.3.22 present a trained word frequency identifying risk aspects from completed rounds of semantic data analysis using training techniques. |
| **Answer**:<br>Risk aspects identified using cyber forensics and semantic data analysis techniques are listed and have sufficient number to adjudicate the sub question. The input is delivered through cyber forensics analysis. Initial analysis of the cyber forensic aspects present artefacts in the form of broad risk aspects. Each round of semantic analysis refinement presents an improved artefact.<br>Therefore, integration and processing utilising both cyber forensic aspects along with semantic analysis aspects allow data refinement to deliver utility and identify an improved artefact output in the form of an instantiation for risk aspect identification. | |

### 7.1.4 RSQ2: Research Sub-Question Two

The following sub-question discriminates which risk aspects in the evidence of Chapter six inform a Capability Maturity Tool for the Internet of Things.

<table>
<tr><td colspan="2"><b>RSQ2</b>: Which risk inputs inform a Capability Maturity Tool for the Internet of Things?</td></tr>
<tr><th>Input</th><th>Description</th></tr>
<tr><td>Input 6.1.1.3, Input 6.2.1.3, and Input 6.3.1.3</td><td>Risk input of meaningful data, selected from Digital Forensic Analysis</td></tr>
<tr><td>Input 6.1.1.3, Input 6.2.1.3, and Input 6.3.1.3</td><td>Risk input of meaningful data, selected from Peer Reviewed Analysis</td></tr>
<tr><td>Input 6.1.1.3, Input 6.2.1.3, and Input 6.3.1.3</td><td>Risk input of meaningful data, selected from Standards and Best Practices</td></tr>
<tr><td>Section 6.1.2, Section 6.2.2, and Section 6.3.2</td><td>Input development to remove non-risk data in the form of StopWords</td></tr>
<tr><td>Section 6.1.3, Section 6.2.3, and Section 6.3.3</td><td>Input training to remove domain specific non-risk data</td></tr>
<tr><td>Section 6.1.4, Section 6.2.4, and Section 6.3.4</td><td>Risk input of trained and conditioned corpora</td></tr>
<tr><td colspan="2"><b>Conclusion:</b><br>The risk inputs that inform the capability maturity tool for the IoT are in three categories. The first is meaningful data, the second is development data, and the third is training data. Each of the three categories contribute to the overall risk impact and identification of risk.</td></tr>
</table>

### 7.1.5 RQ: The Research Question

The research question is designed to locate what factors improve Capability Maturity Risk Modelling for the Internet of Things. The evidence presented in Chapter six is evaluated and presented below.

<table>
<tr><td colspan="2"><b>RQ1</b>: What factors improve Capability Maturity Risk Modelling for the Internet of Things?</td></tr>
<tr><th>Factor</th><th>Description</th></tr>
</table>

| | |
|---|---|
| Identifier: | Factors identifying the overarching domain of the IoT application |
| Domain: | Factors identifying the field of a discipline area |
| Information Links: | Information link factors |
| Information Vocabulary: | Factors selected for domain specific corpora inclusion |
| Cyber Forensic Analysis: | Factors input from legally admissible submissions and peer reviewed cyber forensic analysis |
| Standards and Best Practices: | Factors input from standards documentation, such as NIST FIPS200 and SP800-53. |
| Semantic Analysis: | Data science workflow factors adapted for Natural Language Processing. |
| Cluster Analysis: | Word frequency cluster factors |
| Word Cloud: | Visualization factors |
| Training: | Data Cleaning. Improved risk matrix factors |
| **Conclusion**: The factors that improve capability maturity risk modelling are in three categories. The first is the vocabulary selection, the second is the semantic engine processing, and the third is the cleaning, training and development of the output. These risk factors contribute to the development of the software based, algorithmic solution. | |

## 7.2 SUMMARY OF KEY FINDINGS

DSRM is used as the basis of both methodology identification and evaluation, as discussed in Chapter three. It is adapted to develop a method that includes control steps and an output evaluation process. Section 3.2.1 lists the proposed artefacts that result from the research output. This section discusses the findings as artefact abstraction, in order of process output, (see Figure 3.3,) and each the artefacts are discussed in turn: Section 2.8.2

describes a NIST best practice manual risk identification process, utilizing components of primitives, whereby descriptions, properties, assumptions and general statements are utilized to identify risk. The taxonomic output in Table 2.2, 2.3, 2.4, 2.5 and Table 2.6 show that the manual analysis of sensors, aggregators, clusters, weights and communication channels are cumbersome and the risk factors identified are difficult to integrate into a business area.

The risk contexts identified in Section 2.2 outline concepts of risk and vulnerability aspects of IoT security, all of which are known to network security engineers. However, these factors are complicated when applied to the specific examples given in Section 2.6 which discusses physical and network layer communication of two IoT communication domains, ZigBee and CANbus. A business will employ specific specialist knowledge to determine risk factors for the determination of risk capability maturity. Therefore, as presented in Section 3.1 the experimental research design is developed to assess specific aspects of the Cyber domain, the Internet of Things, and specifically the design of an automated assignment of risk levels. This allows a business decision maker, who is not an expert in either communication domain, to make informed decisions. This research is designed to provide a tool that can be used by company managers who do not have specific domain knowledge.

Thus, the overarching focus of the research design is to determine what factors improve capability maturity risk modelling for the specific domain of the Internet of Things. Table 7.2.1 shows the key findings in each of the segmentations of the design science framework.

***Table 7.2.1: Key findings***

| | |
|---|---|
| **Exaptation** | Known Solution Extended to New Problems |
| **Inference**: | Output forms the initial pilot study and Proof-of-Concept.<br>The inference formed is that there is a link between cyber forensic analysis and IoT risk aspects. |
| **Method**: | Test case manual risk identification. |

| | |
|---|---|
| | Validation identification and process testing control. |
| **Instantiation**: | Semantic Analysis Engine<br>The design and construction of the Semantic Analysis Engine is a system implementation prototype. |
| **Construct**: | The taxonomy output derived from the semantic analysis process, in the form of a construct artefact, informs the following artefact. |
| **Model**: | The Oxford maturity model architecture artefact processes the taxonomy construct to output the final artefact. |
| **Prototype Instantiation**: | The final artefact output is the IoT Risk Maturity Model prototype. |

## 7.3 INTERPRETATION OF KEY FINDINGS

Section 2.4 describes a NIST best practice manual risk identification process, utilizing components of primitives, whereby descriptions, properties, assumptions and general statements are applied to identify risk. The taxonomic output in Table 2.2, 2.3, 2.4, 2.5 and Table 2.6 show that the manual analysis of sensors, aggregators, clusters, weights and communication channels are cumbersome, and the risk factors identified are difficult to integrate into a business process.

The risk contexts identified in Section 2.2 outlined concepts of risk and vulnerability aspects for IoT security, which are all known to network security engineers. However, these factors are complicated when applied to the specific examples given in Section 2.3 which discusses physical and network layer communication of two IoT communication domains, ZigBee and CANbus. A business will employ specific specialist knowledge to determine risk factors to determine risk capability maturity. Therefore, as presented in Section 3.1 the experimental research design is developed to assess specific aspects of the Cyber domain, the Internet of Things, and specifically the design of an automated assignment of risk levels, from the business viewpoint. Thus, the overarching focus of the research design is to determine what factors improve capability maturity risk modelling for the Internet of Things. Each of the key findings is interpreted and discussed for transfer to a business context in the following sub-sections.

### 7.3.1 Exaptation

Effective artefacts exist in related problem areas that may be adapted or, more accurately, exapted to the new problem context. In this space are contributions where design knowledge that already exists in one field is extended or refined so that it can be used in some new application area. This type of research is common in IS, where new technology advances often require new applications and a consequent need to test or refine prior ideas. Often, these new advances open opportunities for the exaptation of theories and artefacts to the new fields. In exaptation research, the researcher needs to demonstrate that the extension of known design knowledge into a new field is nontrivial and interesting. The new field must present some particular challenges that were not present in the field in which the techniques have already been applied. A business expert system fits these requirements.

### 7.3.2 Inference

The inference formed is that there is a link between cyber forensic analysis and the identification of IoT risk aspects. The initial artefact produced from this research, in the form of an inference, is a novel contribution. Inference as an artefact provides initial direction towards a solution to the problem identified. The inference artefact is an output from the problem identified and selected from the literature review undertaken in Chapter two. The problem requires a solution, engendering an initial inference, then calling for a hypothesis. The inference as an artefact provides an opportunity for the researcher to seek evaluation of the overarching conceptual undertakings of the research. The inference that there is a correlation between cyber forensics and risk identification aspects is the conceptual inference to provide a solution to the problem derived from the literature review. Identifying an inference as an artefact provides a foundation to the research and is validated through the process of research proposal and candidate oral (AUT PGR9) confirmation formalities. The testing for correlations between cyber forensics and risk identification aspects form the foundation of the design principles, research design, and design decisions. The inference that an output of a risk maturity model can be formed from analysis of cyber forensic investigations into IoT, is novel and interesting.

### 7.3.3 Method

Chapter two identifies that the problem encompasses theoretical, taxonomical, legal, technical and business domains. The use of design science in this research is to create and design artefacts to provide solutions and evaluate utility of solving those problems.

Section 3.2 indicates that the artefact can combine techniques, procedures, technologies and tools. An artefact in the form of a method is therefore a valuable contributing to research, as the method establishes a process sequence that can be refined or adapted for use in other contexts. Thus, the method artefact is a valid output from the findings presented in Chapter six. The findings from each of the test case scenarios, are produced using the same method, providing evidence that the method itself has validity. The method did not require adjustment for the test cases, which are of widely disparate domains of the IoT discipline, by design. The method artefact is a purposeful, process artefact, as discussed in Section 3.2. The method artefact is the result of a design refinement process, and the evaluation parameter is utility, where the iterative refinement input focused on whether the method worked. Chapter five discusses the requirement to vary the method, demonstrating the utility of the design science process, and allowing the research to develop a final artefact that is functional and effective.

The method artefact is a technical artefact; therefore, performance evaluation provides validity, without social or real-world input. Thus, the validity of the artefact design is demonstrated through the effective application across all three of the case studies. The method artefact is a novel, constructive design, expatriating data science and natural language processing to solve a new problem of risk identification and enumeration, as discussed in Chapter two. The research contribution of the method artefact is prescriptive in the form of techniques and also descriptive providing classification and pattern identification.

### 7.3.4 Instantiation

The Semantic Analysis Engine (SAE) instantiation is designed to be a technical artefact in that there is no human user identified in the semantic analysis component, as described in Section 3.2. The artefact is designed to remove social implications from the use and adoption of the instantiation, presenting an automated autonomous component. The exaptation of Natural Language Processing (NLP) for use in the new application of semantic evaluation for risk analysis presents a novel research contribution. As discussed in Chapter five, the adaptation of an open-source solution presented design efficiencies and resource utility. Counter to the more usual method of NLP, no sentiment appraisal is undertaken. As the findings demonstrate in Section 6.1.1.1, 6.2.1.1, and 6.3.1.1 the SAE instantiation artefact is applicable to each of the three diverse case studies. Though the corpus for each run of the experiment was different, using data input from different sources, and from different dates, the SAE applied the same process and program. The significance of the practical aspects of the SAE are that the concept of the SAE is established, and validation is acquired through utility and application.

However, the initial application of the SAE produced an output that, whilst effective, was not efficient. Several DS evaluation iterations presented individual training components, for each of the three case study applications, and gave a refined artefact output. Data Science workflow integration into DS, adapting Word Cloud to present a visual output of the dataset, provided an effective semantic analysis training solution. The training aspects, as seen in Section 6.1.3, 6.2.3, and 6.3.3 are different for each separate domain, but would only need to be applied once for that specific domain, and subsequent applications within that domain are automated. Because the design and construction of the Semantic Analysis Engine is a system implementation prototype, the artefact designed provides avenues for future research and refinement. The instantiation prototype artefact output from this research provides Proof-of-Concept and presents a novel theoretical exaptation knowledge contribution, which is consistent with Gregor (2013), Hevener (2010), and Peffers (2012).

### 7.3.5 Construct

The Taxonomy construct as an artefact is formed as an integral component of the Semantic Analysis process. The Taxonomy creation, starting with an input of information accumulation as seen in Section 6.1.1.2, 6.2.1.2 and 6.3.1.2, and subjection to a domain relevant term extraction process to output risk attributes. The process is used to present a construct artefact that provides comprehensive as well as concise vocabulary components as seen in Section 6.1.2.1, 6.2.2.1, 6.3.2.1. The construct artefact provides the risk attribution terms of references presenting a solution domain taxonomy construct. The selection of new information input in the form of an ongoing corpora development, offers the ability to refine throughout the design application phases. This affords adaptation for change and process adjustment.

The conditioning and training of the vocabulary components is achieved by manual filtering processes. These processes differentiate between domains, so the unique domain specific features are fairly represented in the datasets. The result is a harmonization of variations that would obstruct clustering processes, and a seamless targeting for maturity elements. The consequence is the delivery of conditioned, domain specific vocabulary components for the model. The process assures that only valuable inputs are processed in the risk maturity creation model.

### 7.3.6 Model

The Risk Maturity artefact delivers a maturity creation model, which consists of abstractions and representations that are utilized to symbolize the associated solution domain. As shown on Section 3.2.2, the risk maturity model artefact provides a set of statements that express relationships within the taxonomy construct. Thus, the risk maturity creation model artefact is a development of the SAE construct artefact, where the model focuses upon utilizing the taxonomy as seen in Section 6.1.3, 6.2.3 and 6.3.3 as input to the risk maturity creation model.

The taxonomy attributes, as output of the SAE process delivers the components utilized to develop the risk maturity model. As presented in Section 3.3.4 and Section 2.6 there are many maturity models available, but very few maturity model creation processes are defined. The Oxford Model has been adapted to provide the risk maturity creation process. Figure 3.8 shows the adaptation of the Oxford maturity model process, utilized to present the risk maturity artefact. Figure 3.8 shows that the identifier, domain and family components of the research risk maturity model adaptation are integral to the text input selection process, the semantic engine process and the taxonomy creation process, as shown in Section 3.1.2. Figure 3.2 shows the four-step practical taxonomy creation steps, which integrate with the Oxford model to output the risk maturity model output artefact. Therefore, the model artefact develops the output from the construct artefact and focuses on utility.

The research contribution of the risk maturity model artefact is the deliverable that expresses the relationship derived from the output of the taxonomy construct artefact. The frequency word count, as shown in 6.1.3, 6.2.3 and 6.3.3, which are then grouped into clusters, as shown in 6.1.4, 6.2.4 and 6.3.4 present a risk maturity output model that is novel and provides business utility.

### 7.3.7 Prototype Instantiation

The final artefact output from this research, is the prototype and system implementation, and represent the culmination of the research outcomes. The prototype instantiation outcome is developed from an analysis of the application of the method, model and construct artefacts. The method artefact presents the process sequence in the form of a constructive design. The construct artefact presents the concise vocabulary components in the form of a taxonomy, as the output of the Semantic Analysis Engine instantiation. The model artefact presents the relationship of the taxonomy process output and the

Oxford maturity creation process to present the risk maturity model. The application of the method, construct and model artefacts form the prototype instantiation.

The prototype instantiation demonstrates a feasible and functional solution to the research problem. The prototype takes a selected information input, in the form of a text vocabulary information accumulation. The text input is then parsed through the semantic engine process to provide a risk maturity output. The prototype has been tested on each of the three test cases defined in Chapters three and four. The output presented in Section 6.1.4, 6.2.4, and 6.3.4 show functionality across each of the widely diverse IoT domains. Chapter four presents an analysis of the different IoT domain considerations, identifying the multifaceted nature of each. The application of the prototype instantiation to each of the three test cases successfully presents a risk maturity analysis. The prototype demonstrates utility and can be seen to be robust and reliable.

## 7.4 VALIDITY AND RELIABILITY

It is important to note that some degree of flexibility may be allowed in judging the degree of evaluation that is needed when new DSR contributions are made—particularly with novel artefacts. Nunamaker (2015) states that a "Proof-of-Concept" is sufficient. When a researcher has expended significant effort in developing an artefact in a project, often with formative testing, the summative testing should not necessarily be expected. It is not expected to be as full or as in-depth in evaluation as is expected in a behavioral research project where the artefact was developed by someone else. The following sub-sections justify the validity and reliability acceptance adopted for this research.

### 7.4.1 Validity

There is a difference between validation, validity, and evaluation as identified by Gregor (2013), Peffers (2012), and Hevner (2010). Each artefact can be evaluated in terms of criteria that may include validity, utility, and efficacy. Within this research validity means that the artefact works and does what it is meant to do. The artefact is shown to be valid by evaluation that it is dependable in operational terms in achieving its goals. The utility criteria assesses whether the achievement of goals has value and can be applied to real-world situations outside the development environment. The evidence of efficacy of the Proof-of-Concept research output, is demonstrated by the successful risk maturity analysis of the three case studies presented in Section 6.1, 6.2, and 6.3. Nunamaker (2015) identifies a process that evaluates whether the Proof-of-Concept design environment can move to the next stage of development. A rigorous design evaluation may draw from many potential techniques, such as analytics, case studies, experiments, or simulations (see Hevner et al. 2004) and naturalistic evaluations.

This research output was not designed to present a fully developed widget. The research process has developed a functioning Proof-of-Concept prototype that is software based. The evaluation of the artefacts individually and the Proof-of-Concept as the research output, utilize analysis techniques using case studies, in keeping with evaluation outlines presented by Hevener (2004) and Nunamaker (2015) and reviewed in Section 3.3.4, 3.5, and 3.6. The techniques evaluate the design process, presenting design utility and efficiency.

Chapter three Nunamaker's (2015) process for determining rigor and relevance in IS research. The evaluation process presented has been integrated to demonstrate that this research output is consistent with Nunamaker's (2015) evaluation process. This research contains aspects of all three stages identified: Proof-of-Concept, Proof-of-Value and Proof-of-Use. However, the research is primarily theoretically based, and therefore provides Proof-of-Concept. The Proof-of-Concept output within this research, is evaluated in terms of functional feasibility of the solution presented. Each of the process steps, and the associated output artefacts demonstrate a solution that is shown to be technically promising and is also advantageous to business risk managers, as the desired target audience. Chapter three examines the key goals of the Proof-of-Concept investigation, which are: functional feasibility, developing a deeper and broader understanding, and to determine initial scholarly knowledge. This will in turn, indicate avenues for further research to develop scholarly theories to explain the artefacts as outcomes of interest.

As outlined in Chapter three, the Proof-of-Concept demonstrated in this research is a rudimentary solution that presents functionality. It tests the functional feasibility of the task to provide a risk maturity model, for the IoT domain. The concept as shown in Sections 2.2, 2.6 and 3.3 involves complex task breakdowns of many stages. Therefore, the Proof-of-Concept nature of the experiment utilizes simple tasks and course-grained actions to present relationships between the system use and output that is of interest to business risk managers. The evaluation determination of this research, as Proof-of-Concept, is to provide an overall determination of whether the design approach is promising. Nunamaker (2015) determines that the endeavors of the research to delineate the goals and barriers to the problem stakeholders, whilst informing and designing indicators and goals, establish a tight connection between relevance and research rigor. The analysis of the findings given in Chapter six have given the researcher a depth of understanding of the phenomena within the problem space of business risk determination within the technically complex domain of the IoT.

The research findings shown in Chapter six present output based on forensic evaluation of three case studies, peer reviewed evaluation, professional standards and best practices. Three case studies are evaluated, and the corpora input in Section 6.1.1.2, 6.2.1.2, and 6.3.1.2, consists of an information accumulation of at least 10, and up to 30 professional analyses of each case study. The information links, (see Figure 3.1), are from legal evidentiary, forensic information, ISO or NIST standards, or best practices accepted by government authorities, such as presented by the National Institute of Justice. Therefore, each corpus presents a reliable information accumulation, (see Figure 3.2), that has followed a text selection process that is both robust and rigorous.

### 7.4.2 Reliability

As outlined in Chapter three, three methods are utilized in this research to evaluate the reliability of the research output presented in Chapter six (and Appendices). The three methods discussed are techniques to evaluate: Prototype, Technical experiment, and Case study, corresponding to Peffer's (2012) outlines. Each of the techniques provide different evaluation vectors and create an evidentiary mesh to scaffold reliable research methods. They each have similarities overlapping concepts and intersecting theoretical supports. The following points provide an overview summary.

- Prototype

  The proposed method to evaluate a prototype analyses the suitability and utility of the implementation of the Proof-of-Concept prototype instantiation artefact. The prototype has been evaluated and shown in Chapter six and discussed within this chapter to provide utility suitable for presenting a risk maturity evaluation. The prototype was applied to three diverse scenarios, and the reliability is demonstrated as utility and suitability through each case study application.

- Technical Experiment

  The proposed method adopted to evaluate a technical experiment is to evaluate the technical performance of the software algorithm implementation. Chapter five presents a method variation to the technical experiment due to poor technical performance. The method variation addressed the technical challenge, and further evaluation provides evidence, in Section 6.1, 6.2 and 6.3 that the Proof-of-Concept algorithm implementation technical performance is suitable. The technical performance provides reliable utility across the application for each of the three diverse scenarios, presenting evidence of reliability throughout the experiment.

- Case Study

  The proposed method that is used to evaluate the Proof-of-Concept prototype is an analysis of the real-world situational application of the algorithm. The case study evaluation analysis provides strong evidence of efficiency and performance, as identified by Peffers (2012) in Chapter three. The algorithm implemented for each of the three case studies present meaningful and accurate information. Case study implementation in this research, presents real-world business scenarios involving situations that have been analyzed in Chapter four, and demonstrates the challenges to business owners seeking to identify risk maturity aspects within the IoT domain. The Proof-of-Concept prototype demonstrates that the algorithm can be utilized by company management, who do not have specific domain knowledge, efficiently and reliably.

## 7.5 IMPLICATIONS

The theoretical foundations of this research, as shown in Section 2.4, 2.5, 2.6 and Section 3.1.1, describe the theoretical requirements for a risk maturity creation process. The integration of taxonomy creation theory (see Section 2.5 and 3.1.1), semantic analysis (see Section 2.5.2) and capability maturity modelling (see Section 2.6) provide the theoretical underpinnings. The novel application of these theories, has been integrated with data science workflow to present the research outcomes. The theoretical steps have been adapted to support the algorithmic process of the practical research undertaken. The results emphasize the theoretical underpinnings, adding to and extending the theory presented in Chapters two and three.

The first theoretical component is aligned with Peffers (2006, 2012) and Gregor (2006, 2013) foundational contributions. Hence, the Design Science Research process begins with an inference, or a conclusion reached on the basis of evidence and reasoning. The inference is not formed before the research problem has been clearly identified, however a definition of the objectives of a potential solution also cannot be defined without an inference being formed. The inference formed during the initial stages of this research arose from the analysis of the research problem. It was the difficult and complex process for a business decision maker to determine risk within the IoT domain. The inference, or conclusion, is that the problem solution lies within a body of domain specific information that is available in a domain specific literature. The reasoning is a logical examination of the standard process of information gathering to learn something new. The evidence is clear, when the manual process of risk identification, as shown in Chapter four is examined. The selection of specific information, followed by a systematic analysis process, will provide an effective solution to the research problem. The extension to theory is, as a product of this research, an inference which is itself an artefact. The inference forms the basis of this research and all subsequent components of the research arise from the inference.

The second theoretical component is the analysis of the desired, effective problem solution. The analysis of the process of taxonomy and maturity modelling theory, as shown in Section 2.8 and Section 2.10, provides evidence that the theory can be adapted to form an algorithmic process. The theoretical underpinnings of the two components, taxonomy and maturity modelling, provide the basis of the initial requirements for text selection, information vocabulary establishment, and the term extraction process steps, as shown in Section 3.1 and Figure 3.2. The theoretical output is aligned to a data science theoretical workflow, which identifies the importance of data selection, collection, and cleaning. Thus, the data science theory, when applied to this research, provides evidence that the highest quality data, in the form of information links, will provide the most efficient and effective artefact output. The extension to the theory is the processes for digital forensic evaluation, and especially analysis that is presented and accepted as legal evidence with a literature base that is high relevant and trustworthy. Therefore, the research presents evidence that the existing theory of data science workflow, adapted for this research, when combined with the theory of taxonomy creation for risk maturity modelling, is enhanced for the legal evidence acceptance requirement.

The third theoretical component contained within this research is exaptation, as defined in Section 3.2, and in accord with Information Science development theory identified by Gregor (2013) and Nunamaker (2015). It is theory presented from a related problem area that is adapted, or exapted into a new problem area. Exaption, as applied within this research, and is utilized to present a novel application of Information Science research theory. It is in the form of data science workflow theory that provides a theoretical problem solution artefact within Design Science Research. The theoretical data science workflow is to gather clean data, apply selected processing techniques, apply text cleaning rounds, then produce document term matrices. The research output presents evidence that data science workflow theory exaptates to Design Science to provide an effective problem solution artefact. The process outputs presented in Section 6.1.1, 6.2.1, and 6.3.1 demonstrate the effectiveness of the application of the design science workflow theory. There are two further theoretical aspects of the IS that have been adopted within this research: providing a visual method output, and utilizing WordCloud (see Section 6.1.2, 6.2.2 and 6.3.2). This visually communicates the results and displays WordCloud to further refine and improve the data output. Thus, the training aspects of this research, as seen in Section 6.1.3, 6.2.3, and 6.3.3, present evidence of the effectiveness of exapting these two data science workflow theoretical inputs. Therefore, the implications for theory are that exaptation of Information Science and data science workflow theory into design science theory are shown to improve the resulting research output artefact.

## 7.6 LIMITATIONS

The artefacts developed and presented in this thesis are tangible process output from each stage of the research design. The artefacts, as listed in Section 3.2.1 are in different forms: inference, method, instantiation, construct, and model. Each artefact is developed with external input to drive a design decision to produce, wherever possible, a better artefact. Therefore, there are a combination of research design principal evaluations, staged partial design decisions, and process output evaluations. The ideal would be to have multiple research teams, where a set of teams provides continuous evaluation of the social / technical integration implications, and a separate set of teams concentrate on enhancing the build of the artefacts. The inclusion of separate teams would reduce the risk of bias that may be present in a small-scale research design and allow streamlined continuous development. However, the nature of the findings presented in this thesis imply a prototype, theoretical design principle, presenting an unfinished output that will provide motivation for future research for artefact development of iterative enhancements, specifications, and wider generalization.

As shown in Section 3.5.4 the problems and opportunities for the stakeholder, require the private and organizational goals, with the economic, political, social, and operational constraints in the environment, and accounts of prior challenges, should also be reported as exploratory findings. As cited in Section 2.5, Nunamaker (2015) shows that quantitative experimental rigor is not useful in Proof-of-Concept research. Proof-of-Concept research is designed to lay a foundation for the next stage of research. Proof-of-Concept researchers can, however, gain advantages over and above limitations that accrue from publishing conceptual papers that do not evaluate real world scenarios. Before a work system solution is developed, for example, researchers can identify if important classes of unsolved problems exist in the field. If they use the disciplines of exploratory research to conduct this work, limitations can be mitigated, and the output designed to set the research agenda for a new branch of scholarly inquiry.

As is outlined in Section 3.5.4, limitations are mitigated through steps to establish value and utility, by deepening scientific understanding of the Proof-of-Concept phenomena and to identify new phenomena, to measure generalizable solutions and to improve functional quality by understanding technical, operational and economic feasibility metrics. A limit was set on the scope of this research to make it a feasible study, but further development and application testing can proceed as further research steps to achieve greater generalizations of the outcomes.

The use of case studies within this research, present domain generalization through selection of three diverse IoT domains, with a time span from 2000 to 2017. This selection of diversity provides an initial sample of IoT technical risk aspects that can be seen to represent many other IoT risk aspect analysis contexts. The Proof-of-Concept software algorithm was applied to each of the case studies, with Section 6.1.5, 6.2.5 and 6.3.5 presenting evidence of an effective IoT risk determination, generalized across IoT domains.

The Proof-of-Concept, as a theoretical construct can be transferred between environments and applied to different domain applications. However, transference to a different domain presents new risk scopes. The level of divergence from the original IoT risk identification environment will present a corresponding greater risk of transference difficulties. The difficulties with domain transference require an assessment of validity as an external reference point, and therefore moving to a different environment may use different reference points, which introduces new risk scope that the research has not been able to treat in the experimental design. However, the careful selection of a vocabulary input through the text selection process shown in Figure 3.2, will enable transfer into new domains. Each has to be retested for validity, performance, and relevance.

## 7.7 CONCLUSION

Chapter seven has discussed the key research findings presented in Chapter six. The Hypotheses have been tested, and the Research sub-questions and the Research question have been answered. The research process has developed seven artefacts as part of the Design Science Research process, each of which not only provide solutions to the problem area defined in Chapter two, but also presented contributions to knowledge. The combined output of this research is an instantiated artefact in the form a functioning Proof-of-Concept prototype that is software based. The evaluation of each artefacts individually and the prototype Proof-of-Concept as a whole, used analysis techniques of the experimental output of three divergent case studies. Chapter seven has shown that the solution objectives defined in Chapter two have been met, through an application of the methods presented in Chapter three. Chapter seven has presented an analysis of the validity and reliability of the prototype, showing that the Proof-of-Concept is valid and reliable. Chapter seven then discussed the implications of three theoretical components of the findings, the initial inference, the defined problem solution and the exaptation use of a data science artefact. The limitations of the research are outlined in the final section of Chapter seven, addressing the risks inherent in transferring knowledge from one domain to a domain that was, for the purposes of this research, out-of-scope. The research question: "What factors improve Capability Maturity Risk Modelling for the Internet of Things?" is answered, and the prototype Proof-of-Concept provides a viable solution to the identified problem area of risk analysis within the IoT domain.

Chapter eight will present a summary of the research, identify the contributions to knowledge, and address recommendations for future research. The research summary will address each of the key findings, assert the knowledge contributions contained within each, and identify how the artefact can be improved through design improvement research and application research. This includes taking the artefact forward and exploring commercialization possibilities. It concludes with the inference that a novel algorithmic approach can be taken to help address cybersecurity risk within the IoT domain, and the next steps for future research.

# CHAPTER 8
# CONCLUSION

## 8.0 INTRODUCTION

This research aimed to develop cybersecurity capability maturity forensic modelling for the Internet of Things. A problem area of difficulty for a business user to enumerate cybersecurity risk within the IoT domain, was adopted to establish the research domain. The research domain informed the research question: *What factors improve Capability Maturity Risk Modelling for the Internet of Things?* Based on established Design Science research methodologies, an output in the form of Proof-of-Concept prototype instantiation artefact, was designed to answer the research question, through experimental testing of two hypotheses: H1- *Risk aspects are identified, using cyber forensic and data analysis techniques* and *H2- The output from H1 informs a Capability Maturity Tool to identify IoT risk.* The findings from the experiment provide evidence of identifiable risk aspects which, in turn, inform the capability maturity tool. The research output delivers seven novel contributions to knowledge as integral components of the Design Science research process.

Chapter eight is structured to first present the research contributions in Section 8.1 and concludes with recommendations for future research and prototype development in Section 8.2.

## 8.1 CONTRIBUTIONS

This research contributes insights into the application of Design Science research methodology, as shown by Gregor (2006, 2013) Hevner (2004, 2010) Nunamaker (2015) and Peffers (2006, 2007, 2012). Therefore, it is in keeping with the overarching Design Science methodology that the research contributions are presented as artefacts. Whilst an artefact is considered something that is an artificially created object, such as a model or instantiation, a method, and software, it also contributes insight into the abstractions used to create it theoretically, or materially. An artefact may have a level of abstraction, an algorithm for example, which may in turn be converted to a material form, in operational software. Whilst a theoretical research output is an abstraction, there is a tangible knowledge addition to the material artefact description. Therefore, both the abstraction / theoretical output and the tangible artefact output from this research, provides important contributions to knowledge.

Table 8.1 presents an itemized overview of the contributions to knowledge from this research. The contributions are presented in turn, with the knowledge contributions outlined. The research contribution is identified as specific instantiations of material / prescriptive knowledge, or abstract design of general / descriptive theoretical knowledge. Each contribution is further elaborated in the following sub-sections.

***Table 8.1: Contributions to knowledge from this research***

| **Artefact** | **Contribution** |
|---|---|
| **Exaptation**<br>(Known solution extended to new problems) | Data Science Workflow methods utilizing Natural Language Processing techniques, adapted to Design Science with an Information Science output. The exaptation research contribution is descriptive in the form of NLP principles, patterns and theories and also prescriptive in the form of algorithms and techniques |
| **Inference**:<br>(Conclusion based on reasoning and evidence) | The inference research contribution is that there is a link between cyber forensic analysis and IoT risk aspects. The inference research contribution is descriptive in the form of observation and classification. |
| **Method**:<br>(Techniques, procedures, technologies and tools) | The research contribution of the method artefact is prescriptive in the form of techniques and also descriptive providing classification and pattern identification. |
| **Instantiation**:<br>(Implementation intended to perform certain tasks) | The design and construction of the Semantic Analysis Engine as a system implementation prototype. The research contribution of the Semantic Analysis Engine is prescriptive knowledge in the form of systems, products and processes. |
| **Construct**: | The taxonomy output derived from the semantic analysis process, in the form of a |

| | |
|---|---|
| (Conceptualization used to specify solutions to identified domain problem) | construct artefact informs the following Model artefact. The research contribution from the construct artefact is prescriptive knowledge, in the form of creation concepts and symbols. |
| **Model**:<br>(Abstraction and representation statements describing tasks expressing relationships among constructs) | The Oxford maturity model architecture artefact processing. The model artefact presents a prescriptive knowledge research contribution in the form of a semantic syntax representation, and also descriptive knowledge in the form of phenomenal observation. |
| **Prototype Instantiation**:<br>(An applied solution designed to have sufficient functionality to test problem solution feasibility) | The final artefact output is the IoT Risk Maturity Model prototype. The research knowledge contribution from the prototype instantiation is prescriptive in the form of a system process implementation. There is also descriptive knowledge contributions in the form of observation, classification and cataloguing, as well as identification of patterns and regularities. |

### 8.1.1 Exaptation

The exaptation, as an artefact output of this research, is that there exists a technique and process in another knowledge domain that can be adapted, or exapted to the problem context of this research. The exaptation output presents prescriptive knowledge contributions of the software, algorithmic techniques, using Data Science workflow and Natural Language Processing (NLP) design knowledge, applied to the problem context. The NLP techniques that already exist in the domain field of Computer Science has been extended and refined in the new application area of autonomous taxonomic creation. In the application of exaptation within this research, the extension of known design knowledge in the form of NLP into the new field of risk identification, is both novel and interesting.

The challenges identified in the problem context, that it is difficult for a nontechnical user to enumerate risk within the IoT domain, is not present in the Computer Science, linguistic research domain. The exaptation of NLP, in the problem context of this research, is a novel use of NLP techniques, where all sentiment is removed from the information accumulation that is processed. There is also a descriptive knowledge contribution component of the exaptation artefact, in the form of phenomenal knowledge, where the observational cataloguing and classification of the NLP technique output is followed by sense-making knowledge in the form of regularities and patterns. The final exapted output from this research presents a multi-faceted, novel knowledge contribution that provides greater understanding of NLP techniques. This includes risk enumeration procedures, and domain specific information accumulation text input selection.

### 8.1.2 Inference

The knowledge contribution presented from the inference artefact shows that there is a link between cyber forensic analysis and the identification of IoT risk aspects. This link is in the form of descriptive knowledge, where the researcher is making sense of observed phenomena. The inference presented in this research is that textual information that is presented as legally acceptable evidence at court, USA Congressional hearings, or presented as US National Transportation Safety Board reports (see Case Study 3) will contain risk enumeration aspects. Therefore, the initial artefact determined the direction of this research, and thus, the inclusion of the inference as an artefact output is a novel knowledge contribution. Inference as a research artefact acknowledges that the inference provides initial direction towards a solution to the problem identified. The problem context may be nascent and as such, not fully formed, however the inference is driving the research of current literature to fully define the problem context. The inference artefact is an output from the initial problem identified and drove selection of the literature reviewed in Chapter two. The problem context requires a solution, informed by an initial inference, then calls for a hypothesis to be tested. The inference as a knowledge contribution artefact provides an opportunity for the researcher to seek evaluation of the overarching conceptual undertakings of the research. The inference that there is a correlation between cyber forensics and risk identification aspects is the descriptive knowledge, conceptual contribution, designed to provide a solution to the fully defined problem derived from the literature review.

Identifying an inference as an artefact provides an observational descriptive knowledge contribution, as the inference is foundational to the research, and in this research, is validated through the process of University research proposal acceptance and candidate confirmation formalities (the PGR9 exam). The testing for correlations between cyber forensics and risk identification aspects form the foundation of the design principles, research design, and design decisions. The inference that an output of a risk maturity model can be formed from analysis of cyber forensic investigations into IoT is novel and interesting. The inclusion of the inference as a research artefact also presents a novel contribution to descriptive knowledge at a theoretical level. The initial declaration of an inference as an artefact provides an insight into the researcher's journey from descriptive observational knowledge, and towards prescriptive knowledge in the form of the construct, the model, the method, and the instantiation artefact output of Design Science Research.

### 8.1.3 Method

A valuable contribution to research is presented by the method artefact where the method establishes a process sequence that can be refined or adapted for use in other contexts. The problem context identified in this research encompasses theoretical, taxonomical, legal, technical and business domains. The purpose of the method artefacts is to provide algorithmic process solutions to the problem context. The method artefact, when applied to each of the widely divergent test case scenarios did not require adjustment for the test cases and produced useful results. This shows that the method itself has validity. The method artefact is a purposeful, process artefact, and is the result of a design refinement process, where the evaluation parameter is utility, and the iterative refinement input focused on whether the method worked. Thus, the validity of the artefact design is demonstrated through the effective application across the three case studies.

The research contribution of the method artefact is prescriptive knowledge of innovative algorithmic software techniques. It uses Data Science and Natural Language Processing to solve the problem of risk identification and enumeration. There are also descriptive knowledge contributions provided through the pattern identification, informing training and conditioning techniques to better present the output classifications. Further knowledge contribution is presented whereby performance evaluation provides assessment of validity, when the method artefact is a Proof-of-Concept prototype. This allows for changes to be made to the research design, quickly and efficiently. The final method artefact from this research demonstrates that the change in design was integrated with minimal disruption to the research direction and flow (Chapter five).

### 8.1.4 Instantiation

The Semantic Analysis Engine (SAE) instantiation artefact, is designed to be a technical artefact in that there is no human user identified in the semantic analysis component, as described in Section 3.2. The artefact is designed to remove social implications from the use and adoption of the instantiation, presenting an automated autonomous component. The exaptation of Natural Language Processing (NLP) for use in the new application of semantic evaluation for risk analysis presents a novel research contribution. As discussed in Chapter five, the adaptation of an open-source solution presented design efficiencies and resource utility. Counter to the more usual method of NLP, no sentiment appraisal is undertaken. As the findings demonstrate in Section 6.1.1.1, 6.2.1.1, and 6.3.1.1, the SAE instantiation artefact is applicable to each of the three diverse case studies, although the corpus for each run of the experiment was different. It used data input from different sources, and from different dates, the SAE applied the same process and program. The significance of the practical testing of the SAE are that the concept of the SAE is established and validation through utility and application.

However, the initial application of the SAE produced an output that, whilst effective, was not efficient. Several DS evaluation iterations had individual training components, for each of the three case study applications, presented a refined artefact output. Data Science workflow integration into DS, adapting Word Cloud to present a visual output of the dataset, provided an effective semantic analysis training solution. The training aspects, as seen in Section 6.1.3, 6.2.3, and 6.3.3 are different for each separate domain, but would only need to be applied once for that specific domain, and subsequent applications within that domain is fully automated. Because the design and construction of the Semantic Analysis Engine is a system implementation prototype, the artefact designed provides avenues for future research and refinement. The instantiation prototype artefact output from this research provides Proof-of-Concept and presents a novel theoretical exaptation knowledge contribution, which is consistent with Gregor (2013), Hevener (2010) and Peffers (2012) requirements.

### 8.1.5 Construct

The Taxonomy construct as an artefact formed as an integral component of the Semantic Analysis process. The Taxonomy creation, starting with an input of information accumulation as seen in Section 6.1.1.2, 6.2.1.2 and 6.3.1.2 is then subjected to a domain relevant term extraction process, to output risk attributes. The process is used to present a construct artefact that provides comprehensive as well as concise vocabulary components as seen in Section 6.1.2.1, 6.2.2.1, 6.3.2.1. The construct artefact provides the risk attribution terms of references presenting a solution domain taxonomy construct. The selection of new information input in the form of an ongoing corpora development offers the ability of refined throughout the design application phases, allowing adaptation for change and process adjustment.

The conditioning and training of the vocabulary components is achieved by manual filtering processes. These processes differentiate between domains, so the unique domain specific features are fairly represented in the datasets. The result is a harmonization of variations that would obstruct clustering processes and targeting of maturity elements. The consequence is the delivery of conditioned, domain specific vocabulary components for the model. The process assures that only valuable inputs are processed in the risk maturity creation model.

### 8.1.6 Model

The Risk Maturity artefact delivers a maturity creation model, which consists of abstractions and representations that are utilized to symbolize the associated solution domain. As shown on Section 3.2.2, the risk maturity model artefact provides a set of statements that express relationships within the taxonomy construct. Thus, the risk maturity creation model artefact is a development of the SAE construct artefact, where the model focuses upon utilizing the taxonomy as seen in Section 6.1.3, 6.2.3 and 6.3.3 as input to the risk maturity creation model.

The taxonomy attributes, as output of the SAE process delivers the components utilized to develop the risk maturity model. As presented in Section 3.3.4 and Chapter two there are many maturity models available, but very few maturity model creation processes are defined. The Oxford Model has been adapted to provide the risk maturity creation process. Figure 3.1 shows the adaptation of the Oxford maturity model process, utilized to present the risk maturity artefact. Figure 3.1 shows that the identifier, domain and family components of the research risk maturity model adaptation are integral to the text input selection process, the semantic engine process and the taxonomy creation process shown in Section 3.1.2. Figure 3.2 shows the four-step practical taxonomy creation steps, which integrate with the Oxford model to output the risk maturity model artefact. Therefore, the model artefact develops the output from the construct artefact, focusing upon utility.

The research contribution of the risk maturity model artefact is the proposition that expresses the relationship derived from the output of the taxonomy construct artefact.

The frequency word count, as shown in 6.1.3, 6.2.3 and 6.3.3, which are then grouped into clusters, as shown in 6.1.4, 6.2.4 and 6.3.4 present a risk maturity output model that is novel and provides business utility.

### 8.1.7 Prototype Instantiation

The final artefact output from this research, is the prototype and system implementation, which is the culmination of the research outcomes. The prototype instantiation outcome is developed from an analysis of the application of the method, model and construct artefacts, as outputs from the research. The method artefact presents the process sequence in the form of a constructive design. The construct artefact presents the concise vocabulary components in the form of a taxonomy, from the output of the Semantic Analysis Engine instanton. The model artefact presents the relationship of the taxonomy process output and the Oxford maturity creation process to present the risk maturity model. The application of the method, construct and model artefacts form the prototype instantiation.

The prototype instantiation demonstrates a feasible and functional solution to the research problem. The prototype takes a selected information input, in the form of a text vocabulary information accumulation. The text input is then parsed through the semantic engine process to provide a risk maturity output. The prototype has been tested on each of the three test cases defined in Chapters three and four. The output presented in Section 6.1.4, 6.2.4, and 6.3.4 show functionality across each of the widely diverse IoT domains. Chapter four presents an analysis of the different IoT domain considerations, identifying the multifaceted nature of each. The application of the prototype instantiation to each of the three test cases successfully presents a risk maturity analysis. The prototype demonstrates in testing utility and can be seen to be robust and reliable.

## 8.2 RECOMMENDATIONS FOR FURTHER RESEARCH

The growth of IoT technology application in the real world is rapid and pervasive. Risk enumeration and determination within the IoT domain is a growing concern to business decision makers. Gaps identified in IoT risk identification methods for business decision-making represent the problem statement addressed in this research. The Proof-of-Concept output presents a solution to the problem statement. However, the nature of the prototype is a theoretical design principle, which is an unfinished output that provides avenues for future research and artefact development. The Proof-of-Concept output of this research is designed to lay a foundation for the next stages of research. The recommendations outlined are focused upon new variations in different domain areas, in terms of Proof-of-Value. Also in future operational feasibility, in terms of Proof-of-Use. Proof-of-Use will recommend further research into wider generalizations into different domain areas such as the finance sector, the health industry and so on. Recommended future research into Proof-of-Value is toward functional development iterative enhancements, investigating specifications for practical use, specifically targeting workplace outcomes and commercialization opportunities.

### 8.2.1 Further Research Recommendations: Proof-of-Value

Future research into Proof-of-Value of the prototype, in keeping with Nunamaker (2015), is to develop improvements of the functional quality of both the processes and the technical components. In this way the research artefacts presented can create value. The research has shown that the prototype has sufficient functionality to solve real problems as demonstrated with the three test cases. However further prototype development research will involve investigation of field trials to establish robust applications of real users performing real work. The development of the prototype into a design suitable for real world applications may require improvisation and workarounds, identified by the key stakeholders. This will involve expert evaluation and iterative input applications to add further input into the design and development process. The process will develop and identify the processes by which the Proof-of-Concept solution can be applied to create value.

The recommended goal of future Proof-of-Value is to investigate solutions to real problems, especially within different conceptual domains. The future research will provide deeper understandings of the phenomena presented as the prototype Proof-of-Concept and to quantify the degree to which the solution can be generalized. Future research of diverse domain applications of this research output will be directed towards measurement of the efficacy of the domain solutions whilst identifying the unintended consequences. The output from research into dissimilar domain generalizations will present system design processes to create value and better assess feasibility issues.

An example of future research into Proof-of-Value would be to investigate the utility of the Proof-of-Concept prototype to enumerate fraud risk within the finance sector. As identified within this research, an initial domain identification, in this example, is the finance sector. The following step is to identify information links as a text input to form an information accumulation. There are numerous cases of financial fraud, malfeasance and illicit financial transactions that have been brought before the courts, where legally acceptable forensic evidence has been presented. There are many sources of financial best practices and widely accepted standards. These would form a valid information accumulation corpus, ready for the term extraction process provided by the Semantic Analysis Engine. The following stages of Taxonomy Creation, and Maturity Modelling would then be tested for generalization through research application. The hypothesis, from this example, is that with the input of a quality corpus, the prototype is agnostic to the specified domain, and therefore generalizable.

Therefore, the nature of the findings presented in this thesis give a prototype, theoretical design principle, with an unfinished output that provides motivation for future research and artefact development. This includes iterative enhancements, specifications, and wider generalization. Research of design improvements involving more complex tasks, analyzing finer grained metrics will present outcomes of interest allowing deeper goal understandings. Further exploratory research of application within different contexts in different conditions can investigate whether the Proof-of-Concept prototype artefact utility is different or similar in different contexts and conditions. Two further areas for future research are quality improvement of the artefact and the other is design feature improvement where Design Science provides iterative improvement of design features that improve performance and improve a conceptual design. Research into Proof-of-Value and Proof-of-Use presents knowledge that innovates fresh design features and domain knowledge.

The research contribution that can be gained through investigations into Proof-of-Value include generalizable requirements and generalizable solutions through development and documentation of exemplar instances of applied solutions. The research output will identify new phenomena of interest, their correlates, and knowledge of new theoretical logic to explain observed phenomena. Research into Proof-of-Value will also present rigorous metrics for solution efficacy and empirical evidence of solution efficacy. Research advantages presented are output in the form of a rich body of explicit and tacit knowledge about the identified problem and solution domains. The knowledge gives dissemination information in the form of a variety of exploratory, theoretical, experimental, and applied science publication.

### 8.2.2 Further Research Recommendations: Proof-of-Use

In keeping with Nunamaker (2015) future research into Proof-of-Use is to investigate the knowledge needed for end users to build instances suitable for the user's problem domain and extending to a generalizable solution. The ideal output of the Proof-of-Use is to present an implementation that is sufficiently robust to run unattended in the workplace, and without requiring product support. The Design Science research process is ideally suited to investigate the integration of output from this research to support every day work processes. Expert input from domain stakeholders informs the direction of future research into Proof-of-Use. The exploratory research of stakeholder's integration experience will give development direction across diverse contexts and conditions. Therefore, the research into Proof-of-Use will present a design theory summarizing the knowledge future implementation requires to successfully integrate their own instances of a generalizable solution. Proof-of-Use research also deepens scholarly understandings of the problem domain and solution spaces.

The recommended goal of future Proof-of-Use research is to integrate exemplar instances of the solution, and instigate experimental research with rigorous, well defined design metrics. The output of the Proof-of-Use research provides community growth of solution practice that is self-supporting and self-sustaining. The research procession will move from theoretical research towards engineering research through definitive problem statements and definitions of key constructs. The theories inform design choices that investigate requirements for multi-domain generalization solution requirements principles in form as well as function. The goal is achieved by implementing rigorous experimental and field tests of design solutions across many domain areas. Thus, the direction of Proof-of-Use future research is to develop in-depth, sophisticated understanding of technical, economic, and operational aspects of the problem and solution spaces. The output will present viable functionality to a wide audience of business users, enumerating risk attributes within a diverse range of subject domains.

The research advantages of the recommended future research of Proof-of-Use is to intentionally develop spheres of knowledge to create business value for problem owners. There is potential of the Proof-of-Use research output to participate in commercialization ventures, which will provide commercialization revenues to deliver resources that advance further research.

# REFERENCES

Abrams, M., & Weiss, J. (2008). Malicious control system cyber security attack case study–Maroochy Water Services, Australia. *McLean, VA: The MITRE Corporation*.

Al-Fuqaha, A., Mohammadi, M., Aledhari, M., Guizani, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys and Tutorials, 17*(4), 2347-2376. https://doi.org/10.1109/COMST.2015.2444095

Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. *Journal of Network and Computer Applications, 88*, 10-28. https://doi.org/https://doi.org/10.1016/j.jnca.2017.04.002

Ali, C. B., Wang, R., & Haddad, H. (2015, 2015//). A Two-Level Keyphrase Extraction Approach. In A. Gelbukh (Chair), *Springer International Publishing.* Symposium conducted at the meeting of the Computational Linguistics and Intelligent Text Processing, Cham.

Aline, D., Daniel Pacheco, L., & Paulo Augusto Cauchick, M. (2015). A Distinctive Analysis of Case Study, Action Research and Design Science Research. *Revista Brasileira de Gestão De Negócios, 17*(56), 1116-1133. https://doi.org/10.7819/rbgn.v17i56.2069

Alturki, A., Gable, G. G., & Bandara, W. (2013). BWW ontology as a lens on IS design theory: extending the design science research roadmap*Springer.* Symposium conducted at the meeting of the International Conference on Design Science Research in Information Systems

Andreas, H., Erik, C., Wei Lee, W., Isam, J., & Stuart, M. (2012). A Unified Approach for Taxonomy-Based Technology Forecasting. In *Business Intelligence Applications and the Web: Models, Systems and Technologies* (pp. 178-197). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-61350-038-5.ch008. https://doi.org/10.4018/978-1-61350-038-5.ch008

Andreas, M. R. (2003). Validity and reliability tests in case study research: a literature review with “““““hands-on””” applications for each research phase. *Qualitative Market Research: An International Journal*(2), 75. https://doi.org/10.1108/13522750310470055

Andruszkiewicz, P., & Hazan, R. (2018, 2018//). Domain Specific Features Driven Information Extraction from Web Pages of Scientific Conferences. In A. Gelbukh (Chair), *Springer International Publishing.* Symposium conducted at the meeting of the Computational Linguistics and Intelligent Text Processing, Cham.

Anthony, S., Karthik, R., Kulathur, S. R., & Gregg, R. M. (2013). Finding Persistent Strong Rules: Using Classification to Improve Association Mining. In *Data Mining: Concepts, Methodologies, Tools, and Applications* (pp. 28-49). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2455-9.ch002. https://doi.org/10.4018/978-1-4666-2455-9.ch002

Antony, B. (2017). *Network packet management optimisation for business forensic readiness*LRetrievedfromhttp://ezproxy.aut.ac.nz/login?url=http://search.ebscohost.com/login.aspx? http://hdl.handle.net/10292/10496

Anusha, K., Senthilkumar, T., & Naik, N. (2017). Development of automatic test script generation (ATSG) tool for active safety software validation*IEEE.* Symposium conducted at the meeting of the Recent Trends in Electronics, Information &

Communication Technology (RTEICT), 2017 2nd IEEE International Conference on

Arunadevi, M., & Perumal, S. K. (2016). Ontology based approach for network security (pp. 573): IEEE.

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks, 54*(15), 2787-2805. https://doi.org/https://doi.org/10.1016/j.comnet.2010.05.010

Balaji, J., Ranjani, P., & Geetha, T. V. (2019). Bootstrapping of Semantic Relation Extraction for a Morphologically Rich Language: Semi-Supervised Learning of Semantic Relations. *International Journal on Semantic Web and Information Systems (IJSWIS), 15*(1), 119-149. https://doi.org/10.4018/IJSWIS.2019010106

Baldwin, T., & Li, Y. (2015). An in-depth analysis of the effect of text normalization in social media. *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, 420-429

Banerjee, T., & Sheth, A. (2017). IoT Quality Control for Data and Application Needs. *IEEE Intelligent Systems, 32*(2), 68-73. https://doi.org/10.1109/MIS.2017.35

Banks, V. A., Plant, K. L., & Stanton, N. A. (2017). Driver error or designer error: Using the Perceptual Cycle Model to explore the circumstances surrounding the fatal Tesla crash on 7th May 2016. *Safety Science*.

Bapna, R., Goes, P., Gupta, A., & Jin, Y. (2004). User Heterogeneity and Its Impact on Electronic Auction Market Design: An Empirical Exploration. *MIS Quarterly, 28*(1), 21-43.

Barnaghi, P., & Sheth, A. (2016). On Searching the Internet of Things: Requirements and Challenges. *IEEE Intelligent Systems, 31*(6), 71-75. https://doi.org/10.1109/MIS.2016.102

Barzegar, M., & Shajari, M. (2018). Attack scenario reconstruction using intrusion semantics. *Expert Systems with Applications, 108*, 119-133. https://doi.org/https://doi.org/10.1016/j.eswa.2018.04.030

Baskerville, R., Bunker, D., Olaisen, J., Pries-Heje, J., Larsen, T. J., & Swanson, E. B. (2014). Diffusion and Innovation Theory: Past, Present, and Future Contributions to Academia and Practice. In B. Bergvall-Kåreborn & P. A. Nielsen (Eds.), *Creating Value for All Through IT: IFIP WG 8.6 International Conference on Transfer and Diffusion of IT, TDIT 2014, Aalborg, Denmark, June 2-4, 2014. Proceedings* (pp. 295-300). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-662-43459-8_18. https://doi.org/10.1007/978-3-662-43459-8_18

Baylon, C., Brunt, R., & Livingstone, D. (2015). *Cyber security at civil nuclear facilities: Understanding the risks*: Chatham House.

Bekara, C. (2014). Security Issues and Challenges for the IoT-based Smart Grid. *Procedia Computer Science, 34*, 532-537. https://doi.org/https://doi.org/10.1016/j.procs.2014.07.064

Bélanger, F., Cefaratti, M., Carte, T., & Markham, S. E. (2014). Multilevel Research in Information Systems: Concepts, Strategies, Problems, and Pitfalls. *Journal of the Association for Information Systems, 15*(9), 614-650.

Belanger, F., & Crossler, R. E. (2011). Privacy in the Digital Age: A Review of Information Privacy Research in Information Systems. *MIS Quarterly, 35*(4), 1017-1041.

Bello, O., Zeadally, S., & Badra, M. (2017). Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT). *Ad Hoc Networks, 57*, 52-62. https://doi.org/https://doi.org/10.1016/j.adhoc.2016.06.010

Benites, F., & Sapozhnikova, E. (2013). Learning Different Concept Hierarchies and the Relations between them from Classified Data. In Data Mining: Concepts, Methodologies, Tools, and Applications (pp. 125-141). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2455-9.ch007. https://doi.org/10.4018/978-1-4666-2455-9.ch007

Bhargav, H. S., Akalwadi, G., & Pujari, N. V. (2016). Application of Blooms Taxonomy in Day-to-Day Examinations*IEEE.* Retrieved from http://ezproxy.aut.ac.nz https://doi.org/10.1109/IACC.2016.157

Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2017). *Network traffic anomaly detection and prevention : concepts, techniques, and tools* [Electronic document]: Cham, Switzerland : Springer, 2017. Retrieved from http://link.springer.com/10.1007/978-3-319-65188-0

Bilobram, G. (2016). Crash Course: How Auto Technology is Changing Claims. (cover story). *Claims, 64*(5), 20-25.

Bordea, G., Buitelaar, P., Faralli, S., & Navigili, R. (2015). SemEval-2015 Task 17: Taxonomy Extraction Evaluation (TExEval)

Borgia, E., Gomes, D. G., Lagesse, B., Lea, R., & Puccinelli, D. (2016). Special issue on "Internet of Things: Research challenges and Solutions". *Computer Communications, 89-90*, 1-4. https://doi.org/https://doi.org/10.1016/j.comcom.2016.04.024

Botta, A., de Donato, W., Persico, V., & Pescapé, A. (2016). Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems, 56*, 684-700. https://doi.org/10.1016/j.future.2015.09.021

Branting, L. K. (2017). Data-centric and logic-based models for automated legal problem solving. *Artificial Intelligence and Law, 25*(1), 5-27. https://doi.org/10.1007/s10506-017-9193-x

Brooks, S. W., Garcia, M. E., Lefkovitz, N. B., Lightman, S., & Nadeau, E. M. (2017). NIST Internal Report NISTIR 8062: An Introduction to Privacy Engineering and Risk Management in Federal Information Systems. *National Institute of Standards and Technology*. https://doi.org/10.6028/NIST.IR.8062

Brumfitt, H. A., Askwith, B., & Zhou, B. (2015). Protecting Future Personal Computing: Challenging Traditional Network Security Models. *2015 IEEE International Conference on Computer & Information Technology; Ubiquitous Computing & Communications; Dependable, Autonomic & Secure Computing; Pervasive Intelligence & Computing*, 1772.

Buccafurri, F., Comi, A., Lax, G., & Rosaci, D. (2016). Experimenting with Certified Reputation in a Competitive Multi-Agent Scenario. *IEEE Intelligent Systems, 31*(1), 48-55. https://doi.org/10.1109/MIS.2015.98

Burton-Jones, A. (2009). Minimizing Method Bias through Programmatic Research. *MIS Quarterly, 33*(3), 445-471.

Burton-Jones, A., & Lee, A. S. (2017). Thinking About Measures and Measurement in Positivist Research: A Proposal for Refocusing on Fundamentals. *Information Systems Research, 28*(3), 451-467. https://doi.org/10.1287/isre.2017.0704

Carina Sofia, A., & Maribel Yasmina, S. (2017). Sentiment Analysis with Text Mining in Contexts of Big Data. *International Journal of Technology and Human Interaction (IJTHI), 13*(3), 47-67. https://doi.org/10.4018/IJTHI.2017070104

Carmela, C., Carlo, M., & Domenico, T. (2006). Metadata, Ontologies, and Information Models for Grid PSE Toolkits Based on Web Services. *International Journal of Web Services Research (IJWSR), 3*(4), 52-72. https://doi.org/10.4018/jwsr.2006100103

Carnegie Mellon University Product Team. (2002). Capability maturity model® integration (CMMI SM), version 1.1. *CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1. 1)*.

Carroll, O. (2017). Challenges in modern digital investigative analysis. *US Att'ys Bull., 65*, 25.

Cheffins, B. R. (2015). The Rise of Corporate Governance in the UK: When and Why. *Current Legal Problems, 68*(1), 387-429. https://doi.org/10.1093/clp/cuv006

Chun-Che, H., & Hao-Syuan, L. (2011). Patent Infringement Risk Analysis Using Rough Set Theory. In *Visual Analytics and Interactive Technologies: Data, Text and Web Mining Applications* (pp. 123-150). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60960-102-7.ch008. https://doi.org/10.4018/978-1-60960-102-7.ch008

Clarke, R., Burton-Jones, A., & Weber, R. (2016). On the Ontological Quality and Logical Quality of Conceptual-Modeling Grammars: The Need for a Dual Perspective. *Information Systems Research, 27*(2), 365-382. https://doi.org/10.1287/isre.2016.0631

Cloutier, M., & Renard, L. (2018). Design Science Research: Issues, Debates and Contributions. *Projectics / Proyéctica / Projectique, 20*(2), 11-16. https://doi.org/10.3917/proj.020.0011

Colby, M., & Tumer, K. (2017). Fitness function shaping in multiagent cooperative coevolutionary algorithms. *Autonomous Agents and Multi-Agent Systems, 31*(2), 179-206. Colby2017. https://doi.org/10.1007/s10458-015-9318-0

Committee on the Judiciary. (2014). Privacy in the digital Age: Preventing data breaches and combating cybercrime. *United States Senate, 113th Congress*.

Culnan, M. J., & Williams, C. C. (2009). How Ethics Can Enhance Organizational Privacy: Lessons from the Choicepoint and TJX Data Breaches. *MIS Quarterly, 33*(4), 673-687.

Cusack, B., & Ward, G. (2018). Points of failure in the ransomware electronic business model.

Cusack, B., Antony, B., Ward, G., & Mody, S. (2017). Assessment of security vulnerabilities in wearable devices. https://doi.org/https://doi.org/10.4225/75/5a84e6c295b44

Dash, S. K., Pakray, P., Porzel, R., Smeddinck, J., Malaka, R., & Gelbukh, A. (2018, 2018//). Designing an Ontology for Physical Exercise Actions. In A. Gelbukh (Chair), *Springer International Publishing.* Symposium conducted at the meeting of the Computational Linguistics and Intelligent Text Processing, Cham.

Datta, S. K., Da Costa, R. P. F., Harri, J., & Bonnet, C. (2016). Integrating connected vehicles in Internet of Things ecosystems: Challenges and solutions. *2016 IEEE 17th International Symposium on A World of Wireless, Mobile & Multimedia Networks (WoWMoM)*, 1.

Daubert v Merrell Dow Pharmaceuticals. (1993). *509 U.S. 579 (1993)*.

De Clercq, S., Bauters, K., Schockaert, S., Mihaylov, M., Nowé, A., & De Cock, M. (2017). Exact and heuristic methods for solving Boolean games. *Autonomous Agents and Multi-Agent Systems, 31*(1), 66-106. De Clercq2017. https://doi.org/10.1007/s10458-015-9313-5

Delone, W. H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: a ten-year update. *Journal of Management Information Systems, 19*(4), 9-30.

De mAAT, E., Winkels, R., & van Engers, T. (2009). Making sense of legal texts. *Formal Linguistics and Law, 212*, 225.

Dharmpal, S. (2017). An Effort to Design an Integrated System to Extract Information Under the Domain of Metaheuristics. *International Journal of Applied Evolutionary Computation (IJAEC), 8*(3), 13-52. https://doi.org/10.4018/IJAEC.2017070102

Ding, W., Yan, Z., & Deng, R. H. (2016). A Survey on Future Internet Security Architectures. *IEEE Access, 4*, 4374-4393. https://doi.org/10.1109/ACCESS.2016.2596705

Durkota, K., Lisý, V., Kiekintveld, C., Bošanský, B., & Pěchouček, M. (2016). Case Studies of Network Defense with Attack Graph Games. *IEEE Intelligent Systems, 31*(5), 24-30. https://doi.org/10.1109/MIS.2016.74

Elliot, S. (2011). Transdisciplinary Perspectives on Environmental Sustainability: A Resource Base and Framework for IT-Enabled Business Transformation. *MIS Quarterly, 35*(1), 197-236.

Emary, I. M. M. E. (2013). Role of Data Mining and Knowledge Discovery in Managing Telecommunication Systems. In *Data Mining: Concepts, Methodologies, Tools, and Applications* (pp. 1591-1606). Hershey, PA, USA: IGI Global. Retrieved from https://doi.org/10.4018/978-1-4666-2455-9.ch083

Ernesto, D., Paolo, C., Angelo, C., Gianluca, E., & Antonio, Z. (2009). KIWI: A Framework for Enabling Semantic Knowledge Management. In *Semantic Knowledge Management: An Ontology-Based Framework* (pp. 1-24). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60566-034-9.ch001. https://doi.org/10.4018/978-1-60566-034-9.ch001

Etuk, A., Norman, T. J., Şensoy, M., & Srivatsa, M. (2017). How to trust a few among many. *Autonomous Agents and Multi-Agent Systems, 31*(3), 531-560. Etuk2017. https://doi.org/10.1007/s10458-016-9337-5

Filho, S. S. S., & Bonacin, R. (2016). *Best Practices in WebQuest Design: Stimulating the Higher Levels of Bloom's Taxonomy*. Conference presented at the meeting of the 2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT) Advanced Learning Technologies (ICALT), Retrieved from http://ezproxy.aut.ac.nz https://doi.org/10.1109/ICALT.2016.29

Finelli, C. J., Borrego, M., & Rasoulifar, G. (2015). Development of a Taxonomy of Keywords for Engineering Education Research. *IEEE Transactions on Education, 58*(4), 219-241. https://doi.org/10.1002/jee.20101

Gan, J., & An, B. (2017). Game-Theoretic Considerations for Optimizing Taxi System Efficiency. *IEEE Intelligent Systems, 32*(3), 46-52. https://doi.org/10.1109/MIS.2017.55

Garima, J., Arun, S., & Sumit Kumar, Y. (2019). Analytical Approach for Predicting Dropouts in Higher Education. *International Journal of Information and Communication Technology Education (IJICTE), 15*(3), 89-102. https://doi.org/10.4018/IJICTE.2019070107

Garud, R., & Kumaraswamy, A. (2005). Vicious and Virtuous Circles in the Management of Knowledge: The Case of Infosys Technologies. *MIS Quarterly, 29*(1), 9-33.

Gass, O., & Maedche, A. (2011). Enabling End-user-driven Data Interoperability-A Design Science Research Project Symposium conducted at the meeting of the AMCIS

Gazis, V. (2017). A Survey of Standards for Machine-to-Machine and the Internet of Things. *IEEE Communications Surveys & Tutorials, 19*(1), 482.

GCSCC (Global Cyber Security Capacity Center) (2014). Cyber Security Capability Maturity Model (CMM). *Oxford Martin School, University of Oxford*. Retrieved from:http://www.intgovforum.org/cms/wks2015/uploads/proposal_background_paper/Cyber-Security-Capacity-Maturity-Model.pdf

Ge, M., Bangui, H., & Buhnova, B. (2018). Big Data for Internet of Things: A Survey. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2018.04.053

Ge, M., Hong, J. B., Guttmann, W., & Kim, D. S. (2017). A framework for automating security analysis of the internet of things. *Journal of Network and Computer Applications, 83*, 12-27. https://doi.org/https://doi.org/10.1016/j.jnca.2017.01.033

Geeta, S. N., & Suresh, N. M. (2017). Survey on Privacy Preserving Association Rule Data Mining. *International Journal of Rough Sets and Data Analysis (IJRSDA), 4*(2), 63-80. https://doi.org/10.4018/IJRSDA.2017040105

Geistfeld, M. A. (2017). A Roadmap for Autonomous Vehicles: State Tort Liability, Automobile Insurance, and Federal Safety Regulation. *California Law Review, 105*(6), 1611-1694. https://doi.org/10.15779/Z38416SZ9R

Genge, B., Graur, F., & Haller, P. (2015). Experimental assessment of network design approaches for protecting industrial control systems. *International Journal of Critical Infrastructure Protection, 11*, 24-38. https://doi.org/10.1016/j.ijcip.2015.07.005

Glaser, B. G. (2016). The Grounded Theory Perspective: Its Origins and Growth. *Grounded Theory Review, 15*(1), 4-9.

Göran, P., Kaj, J. G., & Jonny, K. (2008). Taxonomies of User-Authentication Methods in Computer Networks. In *Information Security and Ethics: Concepts, Methodologies, Tools, and Applications* (pp. 737-760). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-937-3.ch054. https://doi.org/10.4018/978-1-59904-937-3.ch054

Gordon, L. A., Loeb, M. P., & Sohail, T. (2010). Market Value of Voluntary Disclosures Concerning Information Security. *MIS Quarterly, 34*(3), 567-594.

Greenidge, C., & Hadrian, P. (2011). Using an Ontology-Based Framework to Extract External Web Data for the Data Warehouse. In Visual Analytics and Interactive Technologies: Data, Text and Web Mining Applications (pp. 39-59). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60960-102-7.ch003. https://doi.org/10.4018/978-1-60960-102-7.ch003

Greenstein, S., & Zhu, F. (2016). Open Content, Linus' Law, and Neutral Point of View. *Information Systems Research, 27*(3), 618-635. https://doi.org/10.1287/isre.2016.0643

Gregor, S. (2006). The Nature of Theory in Information Systems. *MIS Quarterly, 30*(3), 611-642.

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly, 37*(2).

Gregor, S., & Jones, D. (2007). The Anatomy of a Design Theory. *Journal of the Association for Information Systems, 8*(5), 313-335.

Grzenda, M., Awad, A. I., Furtak, J., & Legierski, J. (2017). *Advances in network systems : architectures, security, and applications* [Electronic document]: Cham : Springer, 2017. Retrieved from http://ezproxy.aut.ac.nz/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat05020a&AN=aut.b19808756&site=eds-live http://ezproxy.aut.ac.nz/login?url=http://link.springer.com/10.1007/978-3-319-44354-6

Gyrard, A., Patel, P., Sheth, A., & Serrano, M. (2016). Building the Web of Knowledge with Smart IoT Applications. *IEEE Intelligent Systems, 31*(5), 83-88. https://doi.org/10.1109/MIS.2016.81

Haberland, V., Miles, S., & Luck, M. (2017). Negotiation strategy for continuous long-term tasks in a grid environment. *Autonomous Agents and Multi-Agent Systems, 31*(1), 130-150. Haberland2017. https://doi.org/10.1007/s10458-015-9316-2

Hansman, S., & Hunt, R. (2005). A taxonomy of network and computer attacks. *Computers & Security, 24*, 31-43. https://doi.org/10.1016/j.cose.2004.06.011

Harland, J., Morley, D. N., Thangarajah, J., & Yorke-Smith, N. (2017). Aborting, suspending, and resuming goals and plans in BDI agents. *Autonomous Agents and Multi-Agent Systems, 31*(2), 288-331. Harland2017. https://doi.org/10.1007/s10458-015-9322-4

Herbig, F. J. (2014). The phronesis of conservation criminology phraseology: A genealogical and dialectical narrative. *Phronimon, 15*(2), 1-17.

Hernandez-Leal, P., Zhan, Y., Taylor, M. E., Sucar, L. E., & Munoz de Cote, E. (2017). Efficiently detecting switches against non-stationary opponents. *Autonomous Agents and Multi-Agent Systems, 31*(4), 767-789. Hernandez-Leal2017. https://doi.org/10.1007/s10458-016-9352-6

Hevner, A., & Chatterjee, S. (2010). Design science research in information systems. In *Design research in information systems* (pp. 9-22): Springer.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Desigb Science in Information Systems Research. *MIS Quarterly, 28*(1), 75-105.

Hicks, D. J. (2018). The Safety of Autonomous Vehicles: Lessons from Philosophy of Science. *IEEE Technology and Society Magazine, Technology and Society Magazine, IEEE, IEEE Technol. Soc. Mag.*(1), 62. https://doi.org/10.1109/MTS.2018.2795123

Hodjat, H., & Maryam, J. (2018). The Role of the Internet of Things in the Improvement and Expansion of Business. *Journal of Organizational and End User Computing (JOEUC), 30*(3), 24-44. https://doi.org/10.4018/JOEUC.2018070102

Hodjat, H., & Reza, M. (2018). Analysis and Evaluation of a Framework for Sampling Database in Recommenders. *Journal of Global Information Management (JGIM), 26*(1), 41-57. https://doi.org/10.4018/JGIM.2018010103

Höller, J., Tsiatsis, V., & Mulligan, C. (2017). Toward a Machine Intelligence Layer for Diverse Industrial IoT Use Cases. *IEEE Intelligent Systems, 32*(4), 64-71. https://doi.org/10.1109/MIS.2017.3121543

Horsmann, T., & Zesch, T. (2016). LTL-UDE@ EmpiriST 2015: tokenization and PoS tagging of social media text. *Proceedings of the 10th web as Corpus workshop*, 120-126

Huang, X., & Ruan, J. (2017). ATL strategic reasoning meets correlated equilibrium*AAAI Press.* Symposium conducted at the meeting of the Proceedings of the 26th International Joint Conference on Artificial Intelligence

Hurlburt, G. F., Voas, J., & Miller, K. W. (2012). The Internet of Things: a reality check. *IT Professional, 14*(3), 56-59.

Ibrahiem Mahmoud Mohamed El, E. (2013). Role of Data Mining and Knowledge Discovery in Managing Telecommunication Systems. In *Data Mining: Concepts, Methodologies, Tools, and Applications* (pp. 1591-1606). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2455-9.ch083. https://doi.org/10.4018/978-1-4666-2455-9.ch083

Ibrahim, G., & Manolya, K. (2013). Data Mining in the Investigation of Money Laundering and Terrorist Financing. In *Data Mining: Concepts, Methodologies, Tools, and Applications* (pp. 2193-2207). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2455-9.ch112. https://doi.org/10.4018/978-1-4666-2455-9.ch112

IEEE Standard for Low-Rate Wireless Networks. (2016). *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, 1-709. https://doi.org/10.1109/IEEESTD.2016.7460875

Iivari, J., Parsons, J., & Wand, Y. (2006). Research in Information Systems Analysis and Design: Introduction to the Special Issue. *Journal of the Association for Information Systems, 7*(8), 509-513.

Ioannidis, J., & Blaze, M. (1993). The architecture and implementation of network-layer security under Unix*Citeseer.* Symposium conducted at the meeting of the Fourth Usenix Security Symposium Proceedings

Iorga, M., Feldman, L., Barton, R., Martin, M. J., Goren, N. S., & Mahmoudi, C. (2018). NIST Special Publication 500-325: Fog Computing Conceptual Model. *National Institute of Standards and Technology*. https://doi.org/10.6028/NIST.SP.500-325

Iqbal, H., Ma, J., Mu, Q., Ramaswamy, V., Raymond, G., Vivanco, D., & Zuena, J. (2017). Augmenting security of internet-of-things using programmable network-centric approaches: a position paper*IEEE.* Symposium conducted at the meeting of the Computer Communication and Networks (ICCCN), 2017 26th International Conference on

Janczewski, L. J., & Ward, G. (2019). IOT: Challenges in Information Security Training.

Johnson, C., Badger, L., Waltermire, D., Snyder, J., & Skorupka, C. (2016). NIST Special Publication 800-150: Guide to Cyber Threat Information Sharing. *NIST, Tech. Rep*.

José, N., & Paula, H. (2013). Optimization of a Hybrid Methodology (CRISP-DM). In *Data Mining: Concepts, Methodologies, Tools, and Applications* (pp. 1998-2020). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2455-9.ch103. https://doi.org/10.4018/978-1-4666-2455-9.ch103

Jusko, J., Rehak, M., Stiborek, J., Kohout, J., & Pevny, T. (2016). Using Behavioral Similarity for Botnet Command-and-Control Discovery. *IEEE Intelligent Systems, 31*(5), 16-22. https://doi.org/10.1109/MIS.2016.88

Kairaldeen, A. R., & Ercan, G. (2015, 2015//). Calculation of Textual Similarity Using Semantic Relatedness Functions. In A. Gelbukh (Chair), *Springer International Publishing.* Symposium conducted at the meeting of the Computational Linguistics and Intelligent Text Processing, Cham.

Kambiz, F., Guangjing, Y., Jing, S., & Satpal Singh, W. (2015). Data Mining for Predicting Pre-diabetes: Comparing Two Approaches. *International Journal of User-Driven Healthcare (IJUDH), 5*(2), 26-46. https://doi.org/10.4018/IJUDH.2015070103

Kaur, H., Chauhan, R., & Alam, M. (2011). An Optimal Categorization of Feature Selection Methods for Knowledge Discovery. *In Visual Analytics and Interactive Technologies: Data, Text and Web Mining Applications* (pp. 94-108). Hershey, PA, USA: IGI Global. https://doi.org/10.4018/978-1-60960-102-7.ch006

Kaye, S., & Kathleen Adair, C. (2015). Demystifying the Delphi Method. In *Research Methods: Concepts, Methodologies, Tools, and Applications* (pp. 84-104). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-7456-1.ch005. https://doi.org/10.4018/978-1-4666-7456-1.ch005

Ketchum, P. (1943). Mathematical Theory of the Differential Analyzer. C. E. Shannon (pp. 63): The National Research Council.

Kenneth David, S. (2015). Risk Management Research Design Ideologies, Strategies, Methods, and Techniques. In *Research Methods: Concepts, Methodologies, Tools, and Applications* (pp. 362-389). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-7456-1.ch017. https://doi.org/10.4018/978-1-4666-7456-1.ch017

Kerrigan, M. (2013). A capability maturity model for digital investigations. *Digital Investigation, 10*(1), 19-33. https://doi.org/https://doi.org/10.1016/j.diin.2013.02.005

Kimberly, L. (2016). Russian Cyberwarfare Taxonomy and Cybersecurity Contradictions between Russia and EU: An Analysis of Management, Strategies, Standards, and Legal Aspects. In *Handbook of Research on Civil Society and National Security in the Era of Cyber Warfare* (pp. 144-161). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-8793-6.ch007. https://doi.org/10.4018/978-1-4666-8793-6.ch007

Kimberly, L. (2019). Russian Cyberwarfare Taxonomy and Cybersecurity Contradictions Between Russia and EU: An Analysis of Management, Strategies, Standards, and Legal Aspects. In *National Security: Breakthroughs in Research and Practice* (pp. 408-425). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-5225-7912-0.ch019. https://doi.org/10.4018/978-1-5225-7912-0.ch019

Klapaftis, I. P., & Manandhar, S. (2010). Taxonomy learning using word sense induction*Association for Computational Linguistics.* Symposium conducted at the meeting of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics

Klapaftis, I. P., & Manandhar, S. (2013). Evaluating Word Sense Induction and Disambiguation Methods. *Language Resources and Evaluation, 47*(3), 579-605. https://doi.org/10.1007/s10579-012-9205-0

Kolias, C., Stavrou, A., Voas, J., Bojanova, I., & Kuhn, R. (2016). Learning Internet-of-Things Security" Hands-On". *IEEE Security & Privacy, 14*(1), 37-46.

Koppenhagen, N., Gaß, O., & Müller, B. (2012). Design Science Research in Action-Anatomy of Success Critical Activities for Rigor and Relevance.

Kovacs, L., & Csizmas, E. (2018). Lightweight ontology in IoT architecture (pp. 1): IEEE.

Kozareva, Z., & Hovy, E. (2010). *A semi-supervised method to learn and construct taxonomies using the web*. presented at the meeting of the Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, Massachusetts.

Lade, P., Ghosh, R., & Srinivasan, S. (2017). Manufacturing Analytics and Industrial Internet of Things. *IEEE Intelligent Systems, 32*(3), 74-79. https://doi.org/10.1109/MIS.2017.49

Lee, C. H., Geng, X., & Raghunathan, S. (2016). Mandatory Standards and Organizational Information Security. *Information Systems Research, 27*(1), 70-86. https://doi.org/10.1287/isre.2015.0607

Li, F., Han, Y., & Jin, C. (2016). Practical access control for sensor networks in the context of the Internet of Things. *Computer Communications, 89-90*, 154-164. https://doi.org/https://doi.org/10.1016/j.comcom.2016.03.007

Liemhetcharat, S., & Veloso, M. (2017). Allocating training instances to learning agents for team formation. *Autonomous Agents and Multi-Agent Systems, 31*(4), 905-940. Liemhetcharat2017. https://doi.org/10.1007/s10458-016-9355-3

Liu, X., Song, Y., Liu, S., & Wang, H. (2012). *Automatic taxonomy construction from keywords*. presented at the meeting of the Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, Beijing, China. https://doi.org/10.1145/2339530.2339754

Loftis, J. D., Forrest, D., Katragadda, S., Spencer, K., Organski, T., Nguyen, C., & Rhee, S. (2018). StormSense: A New Integrated Network of IoT Water Level Sensors in the Smart Cities of Hampton Roads, VA. *Marine Technology Society Journal, 52*(2), 56-67.

Lopez, J., Rios, R., Bao, F., & Wang, G. (2017). Evolving privacy: From sensors to the Internet of Things. *Future Generation Computer Systems, 75*, 46-57. https://doi.org/10.1016/j.future.2017.04.045

Louise, L., & Thomas, M. (2019). Semantic Technologies and Big Data Analytics for Cyber Defence. In *Web Services: Concepts, Methodologies, Tools, and Applications* (pp. 1430-1443). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-5225-7501-6.ch074. https://doi.org/10.4018/978-1-5225-7501-6.ch074

Lutui, R. (2016). A Multidisciplinary Digital Forensic Investigation Model. *Business Horizons*, 59(6), 593-604.

Mack, M. J. (2018). *Security and Threat Analysis of Industrial Control Systems and Applicable Solutions*. Utica College.

Mack, M. J. (2018). Security and threat analysis of industrial control systems and applicable solutions.

Maliah, S., Shani, G., & Stern, R. (2017). Collaborative privacy preserving multi-agent planning. *Autonomous Agents and Multi-Agent Systems, 31*(3), 493-530. Maliah2017. https://doi.org/10.1007/s10458-016-9333-9

Manworren, N., Letwat, J., & Daily, O. (2016). Business Law & Ethics Corner: Why you should care about the Target data breach. *Business Horizons, 59*, 257-266. https://doi.org/10.1016/j.bushor.2016.01.002

Marcello, L. (2009). OntoExtractor: A Tool for Semi-Automatic Generation and Maintenance of Taxonomies from Semi-Structured Documents. In *Semantic Knowledge Management: An Ontology-Based Framework* (pp. 51-73). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60566-034-9.ch003. https://doi.org/10.4018/978-1-60566-034-9.ch003

Marcialis, G. L., Roli, F., Coli, P., & Delogu, G. (2010). A Fingerprint Forensic Tool for Criminal Investigations. *In Handbook of Research on Computational Forensics, Digital Crime, and Investigation: Methods and Solutions* (pp. 23-52). Hershey, PA, USA: IGI Global. https://doi.org/10.4018/978-1-60566-836-9.ch002

Margaret, D. L. (2000). Analyzing Qualitative Data. *Theory Into Practice, 39*(3), 146.

Maria, K. K. (2019). Semantic Intelligence. In *Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction* (pp. 158-167). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-5225-7368-5.ch013. https://doi.org/10.4018/978-1-5225-7368-5.ch013

Mashal, I., Alsaryrah, O., Chung, T.-Y., Yang, C.-Z., Kuo, W.-H., & Agrawal, D. P. (2015). Choices for interaction with things on Internet and underlying issues. *Ad Hoc Networks, 28*, 68-90. https://doi.org/https://doi.org/10.1016/j.adhoc.2014.12.006

Maxwell, J. C., Antón, A. I., Swire, P., Riaz, M., & McCraw, C. M. (2012). A legal cross-references taxonomy for reasoning about compliance requirements. *Requirements Engineering, 17*(2), 99-115.

Mayer, S., Hodges, J., Yu, D., Kritzler, M., & Michahelles, F. (2017). An Open Semantic Framework for the Industrial Internet of Things. *IEEE Intelligent Systems, 32*(1), 96-101. https://doi.org/10.1109/MIS.2017.9

McKinney, E. H., & Yoos, C. J. (2010). Information About Information: A Taxonomy of Views. *MIS Quarterly, 34*(2), 329-344.

Menard, P., Bott, G. J., & Crossler, R. E. (2017). User Motivations in Protecting Information Security: Protection Motivation Theory Versus Self-Determination Theory. *Journal of Management Information Systems, 34*(4), 1203-1230. https://doi.org/10.1080/07421222.2017.1394083

Mieke, J., Nadine, L., & Koen, V. (2013). Data Mining and Economic Crime Risk Management. In *Data Mining: Concepts, Methodologies, Tools, and Applications* (pp. 1664-1686). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2455-9.ch087. https://doi.org/10.4018/978-1-4666-2455-9.ch087

Mineraud, J., Mazhelis, O., Su, X., & Tarkoma, S. (2016). A gap analysis of Internet-of-Things platforms. *Computer Communications, 89-90*, 5-16. https://doi.org/https://doi.org/10.1016/j.comcom.2016.03.015

Minerva, R., Biru, A., & Rotondi, D. (2015a). *Towards a definition of the internet of things*: Institute of Electrical and Electronic Engineers (IEEE). Retrieved from https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf

Minerva, R., Biru, A., & Rotondi, D. (2015b). Towards a definition of the Internet of Things (IoT). *IEEE Internet Initiative, 1.*

Moody, G. D., Kirsch, L. J., Slaughter, S. A., Dunn, B. K., & Weng, Q. (2016). Facilitating the Transformational: An Exploration of Control in Cyberinfrastructure Projects and the Discovery of Field Control. *Information Systems Research, 27*(2), 324-346. https://doi.org/10.1287/isre.2016.0619

Mustard, S. (2005). Security of distributed control systems: The concern increases. *IEE Computing and Control Engineering, 16*(6), 19-25. https://doi.org/10.1049/cce:20050605

Mustard, S. (2006). The celebrated maroochy water attack. *Computing & Control Engineering Journal, 16*(6), 24-25.

National Transportation Safety Board. (2017). NTSB/HAR-17/02 Collisoion between a car operating with automated vehicle control systems and a tractor-semitrailer truck near Williston, Florida, May 7, 2016. Washington D.C.: NTSB. Retrieved from https://www.ntsb.gov/investigations/accidentreports/reports/har1702.pdf

Netolicka, J., & Simonova, I. (2017). *SAMR Model and Bloom's Digital Taxonomy Applied in Blended Learning/Teaching of General English and ESP.* Conference presented at the meeting of the 2017 International Symposium on Educational Technology (ISET), Retrieved from http://ezproxy.aut.ac.nz https://doi.org/10.1109/ISET.2017.68

NIST, & Barrett, M. P. (2018). *Framework for Improving Critical Infrastructure Cybersecurity Version 1.1.*

NIST, & Voas, J. (2016). Networks of 'Things'(NIST Special Publication 800-183). *National Institute of Standards and Technology, 30*, 30.

NIST 200. (2006). Minimum Security Requirements for Federal Information and Information Systems. https://doi.org/https://doi.org/10.6028/NIST.FIPS.200

NIST. (2014). Improving critical infrasructure cybersecurity executive order 13636. Retrieved from www.nist.gov/itl/upload/preliminary-cybersecurity-framework.pdf

NIST 800-53. (2020). Security and Privacy Controls for Information Systems and Organizations. https://doi.org/https://doi.org/10.6028/NIST.SP.800-53r5

Nunamaker, J. F., Briggs, R. O., Derrick, D. C., & Schwabe, G. (2015). The Last Research Mile: Achieving Both Rigor and Relevance in Information Systems Research. *Journal of Management Information Systems, 32*(3), 10-47. https://doi.org/10.1080/07421222.2015.1094961

Nunamaker Jr, J. F., Chen, M., & Purdin, T. D. (1990). Systems development in information systems research. *Journal of Management Information Systems, 7*(3), 89-106.

Ö, K., Ajmeri, N., & Singh, M. P. (2016). Revani: Revising and Verifying Normative Specifications for Privacy. *IEEE Intelligent Systems, 31*(5), 8-15. https://doi.org/10.1109/MIS.2016.89

Offermann, P., Levina, O., Schönherr, M., & Bub, U. (2009). *Outline of a design science research process*. https://doi.org/10.1145/1555619.1555629

Olaronke, O. F. (2018). Indexing and Abstracting as Tools for Information Retrieval in Digital Libraries: A Review of Literature. In *Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications* (pp. 905-927). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-5225-5191-1.ch039. https://doi.org/10.4018/978-1-5225-5191-1.ch039

Oleg, O. (2011). Ensembles of Classifiers. In *Feature Selection and Ensemble Methods for Bioinformatics: Algorithmic Classification and Implementations* (pp. 252-259). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60960-557-5.ch016. https://doi.org/10.4018/978-1-60960-557-5.ch016

Oliveira, T., Satoh, K., Novais, P., Neves, J., & Hosobe, H. (2017). A dynamic default revision mechanism for speculative computation. *Autonomous Agents and Multi-Agent Systems, 31*(3), 656-695. Oliveira2017. https://doi.org/10.1007/s10458-016-9341-9

Olivier, F., Carlos, G., & Florent, N. (2015). New Security Architecture for IoT Network. *Procedia Computer Science, 52*, 1028-1033. https://doi.org/10.1016/j.procs.2015.05.099

Onofrejová, D., Onofrej, P., & Šimšík, D. (2014). Model of Production Environment Controlled With Intelligent Systems. *Procedia Engineering, 96*, 330-337. https://doi.org/10.1016/j.proeng.2014.12.128

Padgette, J., Bahr, J., Batra, M., Holtman, M., Smithbey, R., Chen, L., & Scarfone, K. (2017). Guide to bluetooth security. *NIST Special Publication, 800*, 121. https://doi.org/10.6028/NIST.SP.800-121r2

Panella, A., & Gmytrasiewicz, P. (2017). Interactive POMDPs with finite-state models of other agents. *Autonomous Agents and Multi-Agent Systems, 31*(4), 861-904. Panella2017. https://doi.org/10.1007/s10458-016-9359-z

Patel, P., Ali, M. I., & Sheth, A. (2017). On Using the Intelligent Edge for IoT Analytics. *IEEE Intelligent Systems, 32*(5), 64-69. https://doi.org/10.1109/MIS.2017.3711653

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). The capability maturity model for software. *Software engineering project management, 10*, 1-26.

Pavlou, P. A. (2011). State of the Information Privacy Literature: Where are We Now And Where Should We Go? *MIS Quarterly, 35*(4), 977-988.

Peffers, K., Rothenberger, M., Tuunanen, T., & Vaezi, R. (2012). Design science research evaluation*Springer*. Symposium conducted at the meeting of the International Conference on Design Science Research in Information Systems

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). The design science research process: a model for producing and presenting information systems research*sn*. Symposium conducted at the meeting of the Proceedings of the first international conference on design science research in information systems and technology (DESRIST 2006)

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management*

*Information Systems, 24*(3), 45-77. https://doi.org/10.2753/MIS0742-1222240302

Pourmirza, S., Peters, S., Dijkman, R., & Grefen, P. (2017). A systematic literature review on the architecture of business process management systems. *Information Systems, 66*, 43-58. https://doi.org/https://doi.org/10.1016/j.is.2017.01.007

Prakash, B. A. (2016). Prediction Using Propagation: From Flu Trends to Cybersecurity. *IEEE Intelligent Systems, 31*(1), 84-88. https://doi.org/10.1109/MIS.2016.1

Pronk, T. E., Pimentel, A. D., Roos, M., & Breit, T. M. (2007). Taking the example of computer systems engineering for the analysis of biological cell systems. *BioSystems, 90*, 623-635. https://doi.org/10.1016/j.biosystems.2007.02.002

Pulkkis, G., Grahn, K., J, & Karlsson, J. (2008). Taxonomies of User-Authentication Methods in Computer Networks. In Information Security and Ethics: Concepts, Methodologies, Tools, and Applications (pp. 737-760). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-937-3.ch054. https://doi.org/10.4018/978-1-59904-937-3.ch054

Raichelson, L., Soffer, P., & Verbeek, E. (2017). Merging event logs: Combining granularity levels for process flow analysis. *Information Systems, 71*, 211-227. https://doi.org/https://doi.org/10.1016/j.is.2017.08.010

Rasekh, A., Hassanzadeh, A., Mulchandani, S., Modi, S., & Banks, M. K. (2016). Smart Water Networks and Cyber Security. *Journal of Water Resources Planning and Management, 142*(7), 01816004. https://doi.org/doi:10.1061/(ASCE)WR.1943-5452.0000646

Ray, P. P. (2016). A survey on Internet of Things architectures. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2016.10.003

Rhee, S. (2016). Catalyzing the internet of things and smart cities: Global city teams challenge*IEEE.* Symposium conducted at the meeting of the Science of Smart City Operations and Platforms Engineering (SCOPE) in partnership with Global City Teams Challenge (GCTC)(SCOPE-GCTC), 2016 1st International Workshop on

Riahi Sfar, A., Natalizio, E., Challal, Y., & Chtourou, Z. (2018). A roadmap for security challenges in the Internet of Things. *Digital Communications and Networks, 4*, 118-137. https://doi.org/10.1016/j.dcan.2017.04.003

Richard, O., Marco, R. S., & Sietse, O. (2019). 3PM Revisited: Dissecting the Three Phases Method for Outsourcing Knowledge Discovery. *International Journal of Business Intelligence Research (IJBIR), 10*(1), 80-93. https://doi.org/10.4018/IJBIR.2019010105

Riley, M., Elgin, B., Lawrence, D., & Matlack, C. (2014). Missed Alarms and 40 Million Stolen Credit Card Numbers: How Target Blew It. *Bloomberg.com*, 1-1.

Ross, R., McEvelley, M., & Oren, J. (2016). NIST special Publication 800-160 Systems Security Engineering-Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems. *National Institute of Standards and Technology*.

Ross, R. S., Feldman, L., & Witte, G. A. (2016). *Rethinking Security through Systems Security Engineering*.

Rowley, J., Liu, A., Sandry, S., Gross, J., Salvador, M., Anton, C., & Fleming, C. (2018). Examining the driverless future: An analysis of human-caused vehicle accidents and development of an autonomous vehicle communication testbed (pp. 58): IEEE.

Ruiz, M., Costal, D., España, S., Franch, X., & Pastor, Ó. (2015). GoBIS: An integrated framework to analyse the goal and business process perspectives in information systems. *Information Systems, 53*, 330-345. https://doi.org/10.1016/j.is.2015.03.007

Saarikko, T., Westergren, U. H., & Blomquist, T. (2017). The Internet of Things: Are you ready for what's coming? *Business Horizons, 60*(5), 667-676. https://doi.org/https://doi.org/10.1016/j.bushor.2017.05.010

Sadeghian, A., Sundaram, L., Wang, D. Z., Hamilton, W. F., Branting, K., & Pfeifer, C. (2018). Automatic semantic edge labeling over legal citation graphs. *Artificial Intelligence and Law, 26*(2), 127-144. https://doi.org/10.1007/s10506-018-9217-1

Sajjid, S., & Yousaf, M. (2014). Security analysis of IEEE 802.15.4 MAC in the context of Internet of Things (IoT) (pp. 9): IEEE.

Sandro, B., Lucile, S., Ludovic, J., & Bruno, F. (2017). Multidimensional Model Design using Data Mining: A Rapid Prototyping Methodology. *International Journal of Data Warehousing and Mining (IJDWM), 13*(1), 1-35. https://doi.org/10.4018/IJDWM.2017010101

Sang, E. T. K., Hofmann, K., & de Rijke, M. (2011). Extraction of Hypernymy Information from Text∗ [Sang2011]. In A. van den Bosch & G. Bouma (Eds.), *Interactive Multi-modal Question-Answering* (pp. 223-245). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-17525-1_10. https://doi.org/10.1007/978-3-642-17525-1_10

Serban, A. C., Poll, E., & Visser, J. (2018). Tactical Safety Reasoning. A Case for Autonomous Vehicles (pp. 1): IEEE.

Shahzad, A., Kim, Y. G., & Elgamoudi, A. (2017, 13-15 Feb. 2017). Secure IoT Platform for Industrial Control Systems Symposium conducted at the meeting of the 2017 International Conference on Platform Technology and Service (PlatCon) https://doi.org/10.1109/PlatCon.2017.7883726

Shalin, H.-J. (2013). Structuring and Facilitating Online Learning through Learning/Course Management Systems. In *Data Mining: Concepts, Methodologies, Tools, and Applications* (pp. 1358-1375). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2455-9.ch070. https://doi.org/10.4018/978-1-4666-2455-9.ch070

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*, 623.

Sheth, A. (2016). Internet of Things to Smart IoT Through Semantic, Cognitive, and Perceptual Computing. *IEEE Intelligent Systems, 31*(2), 108-112. https://doi.org/10.1109/MIS.2016.34

Shigarov, A. O., & Mikhailov, A. A. (2017). Rule-based spreadsheet data transformation from arbitrary to relational tables. *Information Systems, 71*, 123-136. https://doi.org/https://doi.org/10.1016/j.is.2017.08.004

Shouhong, W. (1996). Toward Formalized Object-Oriented Management Information Systems Analysis. *Journal of Management Information Systems*(4), 117.

Shu, X., Tian, K., Ciambrone, A., & Yao, D. (2017). Breaking the target: An analysis of target data breach and lessons learned. arXiv preprint arXiv:1701.04940.

Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks, 76*, 146-164. https://doi.org/https://doi.org/10.1016/j.comnet.2014.11.008

Sicari, S., Rizzardi, A., Miorandi, D., & Coen-Porisini, A. (2017). Security towards the edge: Sticky policy enforcement for networked smart objects. *Information Systems, 71*, 78-89. https://doi.org/https://doi.org/10.1016/j.is.2017.07.006

Singh, V., Dwarakanath, T., Haribabu, P., & Babu, S. C. (2017). IoT standardization efforts — An analysis (pp. 1083): IEEE.

Slay, J., & Miller, M. (2007). *Lessons Learned from the Maroochy Water Breach* (Vol. 253). https://doi.org/10.1007/978-0-387-75462-8_6

Smith, S., Winchester, D., Bunker, D., & Jamieson, R. (2010). Circuits of Power: A Study of Mandated Compliance to an Information Systems Security "De Jure" Standard in a Government Organization. *MIS Quarterly, 34*(3), 463-486.

Soltani, S., & Seno, S. A. H. (2017, 26-27 Oct. 2017). A survey on digital evidence collection and analysis Symposium conducted at the meeting of the 2017 7th International Conference on Computer and Knowledge Engineering (ICCKE) https://doi.org/10.1109/ICCKE.2017.8167885

Somanchi, S., & Neill, D. B. (2017). Graph Structure Learning from Unlabeled Data for Early Outbreak Detection. *IEEE Intelligent Systems, 32*(2), 80-84. https://doi.org/10.1109/MIS.2017.25

Son, J.-Y., & Kim, S. S. (2008). Internet Users' Information Privacy-Protective Responses: A Taxonomy and a Nomological Model. *MIS Quarterly, 32*(3), 503-529.

Stanković, R., Štula, M., & Maras, J. (2017). Evaluating fault tolerance approaches in multi-agent systems. *Autonomous Agents and Multi-Agent Systems, 31*(1), 151-177. Stanković2017. https://doi.org/10.1007/s10458-015-9320-6

Stoneburner, G., Goguen, A. Y., & Feringa, A. (2002). NIST Special Publication 800-30: Risk management guide for information technology systems. *National Institute of Standards and Technology*.

Stouffer, K., Lightman, S., Pillitteri, V., Abrams, M., & Hahn, A. (2014). NIST special publication 800-82: Guide to industrial control systems (ICS) security. *National Institute of Standards and Technology*.

Strasser, A. (2017). Delphi Method Variants in Information Systems Research: Taxonomy Development and Application. *Electronic Journal of Business Research Methods, 15*(2), 120-133.

Strong, D. M., & Volkoff, O. (2010). Understanding Organization&#x2014;Enterprise System Fit: A Path to Theorizing the Information Technology Artifact. *MIS Quarterly, 34*(4), 731-756.

Suo, H., Wan, J., Zou, C., & Liu, J. (2012, 23-25 March 2012). Security in the Internet of Things: A Review Symposium conducted at the meeting of the 2012 International Conference on Computer Science and Electronics Engineering https://doi.org/10.1109/ICCSEE.2012.373

Supreme Court of Queensland R v Boden [2002] QCA 164 (2002).

Suresh, P., Daniel, J., Parthasarathy, V., & Aswathy, R. (2014). A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment (pp. 1): IEEE.

Taccari, L., Sambo, F., Bravi, L., Salti, S., Sarti, L., Simoncini, M., & Lori, A. (2018). Classification of Crash and Near-Crash Events from Dashcam Videos and Telematics (pp. 2460): IEEE.

Tewari, A., & Gupta, B. B. (2018). Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2018.04.027

Thangaraj, M., & Sujatha, G. (2014). An architectural design for effective information retrieval in semantic web. *Expert Systems with Applications, 41*(18), 8225-8233. https://doi.org/https://doi.org/10.1016/j.eswa.2014.07.017

Thomas, M. C., & Chris, D. (2008). An Overview of Electronic Attacks. In *Information Security and Ethics: Concepts, Methodologies, Tools, and Applications* (pp. 532-553). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-937-3.ch041. https://doi.org/10.4018/978-1-59904-937-3.ch041

Thuraisingham, B., Tsybulnik, N., & Alam, A. (2008). Administering the Semantic Web: Confidentiality, Privacy, and Trust Management. In Information Security and Ethics: Concepts, Methodologies, Tools, and Applications (pp. 72-88). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-937-3.ch005. https://doi.org/10.4018/978-1-59904-937-3.ch005

Tompkins, A. (2017). Science in the courtroom: is there, and should there, be a better way? *Australian Journal of Forensic Sciences, 49*(5), 579-588. https://doi.org/10.1080/00450618.2016.1236293

Trappey, A. J. C., Trappey, C. V., Hareesh Govindarajan, U., Chuang, A. C., & Sun, J. J. (2017). A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0. *Advanced Engineering Informatics, 33*, 208-229. https://doi.org/https://doi.org/10.1016/j.aei.2016.11.007

Tri, W. (2011). From Data to Knowledge: Data Mining. In *Visual Analytics and Interactive Technologies: Data, Text and Web Mining Applications* (pp. 109-121). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60960-102-7.ch007. https://doi.org/10.4018/978-1-60960-102-7.ch007

Troels, A., & Henrik, B. (2008). Query Expansion by Taxonomy. In *Handbook of Research on Fuzzy Information Processing in Databases* (pp. 325-349). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-853-6.ch013. https://doi.org/10.4018/978-1-59904-853-6.ch013

Usman, M., Britto, R., Borstler, J., & Mendes, E. (2017). Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method (Vol. 85, pp. 43-59).

Vaishnavi, V. K., & Kuechler, W. (2015). *Design science research methods and patterns: innovating information and communication technology*: Crc Press.

Venable, J., & Baskerville, R. (2012). Eating our own Cooking: Toward a More Rigorous Design Science of Research Methods. *Electronic Journal of Business Research Methods, 10*(2), 141-153.

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: a Framework for Evaluation in Design Science Research. *European Journal of Information Systems, 25*(1), 77-89. https://doi.org/10.1057/ejis.2014.36

Vilarinho, C., Tavares, J. P., & Rossetti, R. J. F. (2016). Design of a Multiagent System for Real-Time Traffic Control. *IEEE Intelligent Systems, 31*(4), 68-80. https://doi.org/10.1109/MIS.2016.66

Voas, J. (2016). Demystifying the Internet of Things. *Computer, 49*(6), 80-83. https://doi.org/10.1109/MC.2016.162

Voas, J., Feldman, L., & Witte, G. ITL Bulletin for September 2016.

Wang, H., Huang, Y., Khajepour, A., Liu, T., Qin, Y., & Zhang, Y. (2018). Local Path Planning for Autonomous Vehicles: Crash Mitigation (pp. 1602): IEEE.

Weber, R. H., & Studer, E. (2016). Cybersecurity in the Internet of Things: Legal aspects. *Computer Law & Security Review, 32*(5), 715-728. https://doi.org/https://doi.org/10.1016/j.clsr.2016.07.002

Wei, J. (2012). Survey of network and computer attack taxonomy (pp. 294): IEEE.

Weiss, M., Eidson, J., Barry, C., Broman, D., Goldin, L., Iannucci, B., & Stanton, K. (2015). *Time-aware applications, computers, and communication systems (TAACCS)*: NIST.

Wolf, M., & Serpanos, D. (2018). Safety and Security in Cyber-Physical Systems and Internet-of-Things Systems. *Proceedings of the IEEE, 106*(1), 9.

Xiao, M., Xiatian, D., & Hang, L. (2017). One resistor and two capacitors: An electrical engineer's simple view of a biological cell (pp. 1): IEEE.

Yan, Z., Zhang, P., & Vasilakos, A. V. (2014). A survey on trust management for Internet of Things. *Journal of Network and Computer Applications, 42*, 120-134. https://doi.org/10.1016/j.jnca.2014.01.014

Yasuhiro, Y., Kanji, K., & Sachio, H. (2013). Text Mining for Analysis of Interviews and Questionnaires. In *Data Mining: Concepts, Methodologies, Tools, and Applications* (pp. 1390-1406). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2455-9.ch072. https://doi.org/10.4018/978-1-4666-2455-9.ch072

Yin, R. K. (2011). *Applications of case study research*: sage.

Yin, R. K. (2014). *Case study research : design and methods* [Bibliographies Non-fiction]: Los Angeles : SAGE, [2014]5th edition. Retrieved from http://ezproxy.aut.ac.nz/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat05020a&AN=aut.b13007440&site=eds-live

Yin, R. K. (2016). *Qualitative research from start to finish* [Electronic document]: New York : Guilford Press, [2016]Second edition. Retrieved from http://ezproxy.aut.ac.nz/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat05020a&AN=aut.b14226479&site=eds-live

http://ezproxy.aut.ac.nz/login?url=http://ebookcentral.proquest.com/lib/aut/detail.action?docID=2008479

Ying, L., Han Tong, L., & Wen Feng, L. (2008). Deriving Taxonomy from Documents at Sentence Level. In *Emerging Technologies of Text Mining: Techniques and Applications* (pp. 99-119). Hershey, PA, USA: IGI Global. Retrieved from http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-373-9.ch005. https://doi.org/10.4018/978-1-59904-373-9.ch005

Zafar, B., Cochez, M., & Qamar, U. (2016, 19-21 Dec. 2016). Using Distributional Semantics for Automatic Taxonomy Induction Symposium conducted at the meeting of the 2016 International Conference on Frontiers of Information Technology (FIT) https://doi.org/10.1109/FIT.2016.070

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications, 84*, 25-37. https://doi.org/https://doi.org/10.1016/j.jnca.2017.02.009

Zhang, M., & Gable, G. G. (2017). A Systematic Framework for Multilevel Theorizing in Information Systems Research. *Information Systems Research, 28*(2), 203-224. https://doi.org/10.1287/isre.2017.0690

# APPENDIX

## APPENDIX A INFORMATION SERVICES AND PROTOCOLS

Appendix A reviews Information Service Models and Protocols that impact and shape the IoT environment. The following four models are defined and reviewed:

- A.1 Internet Services and Protocols
- A.2 Communication Reference Models
- A.3 TCP/IP Stack Protocols
- A.4 IoT Reference Models

## APPENDIX A.1 INTERNET SERVICES AND PROTOCOLS

Although protocols and services are different concepts in computer networks, they must interact in order to be functional.

- Service

A service is a set of functions that the network layer of the OSI stack (see Appendix A.2) can provide to a layer above at each layer in the model. The service does not specify any operational implementation information, but rather defines operations and states for the layer that is ready to provide the service.

- Protocol

A protocol is a set of rules used to govern a message’s meaning and format (a frame on Ethernet) that are exchanged between peers. The process allows entities to utilize protocols to then implement their services. The protocol used can be changed as long as the service states and operations are ensured.

### Services

There are two types of services, connection-oriented and connectionless services.

#### *Connection Oriented Service*

A sequence of operations are undertaken in order to provide a connection oriented service. The sequence is to first establish a connection, then to utilize the connection formed, and finally to release the connection. Upon establishing a connection, the sender and receiver of data can negotiate the parameters such as quality of service and size of the transmission units for the communication provided by the service.

*Connectionless Service*

Each message from source to receiver is routed independently across the network. Connectionless services provide fast data transmission as there is no negotiation process involved during connection. The drawbacks of the connectionless service provision can be that packets can be received out of order, be corrupted, or even missing entirely. There is no error checking or flow control provided with connectionless service provision.

## APPENDIX A.2 COMMUNICATION REFERENCE MODELS

Formal and logically structured reference models are invaluable when discussing networking data exchange processes. Reference models define the interaction and functionality of network devices and software. Two popular reference models are utilized, the Open System Interconnect (OSI) reference model, designed by the International Organization for Standardization (ISO) and the TCP/IP reference model designed for the ARPANET research project on network interconnectivity.

### OSI REFERENCE MODEL

A communication system that overrides national and global boundaries is necessary when developing worldwide compatibility. As a result, the ISO international Organization for Standardization worked to develop the Open Systems Interconnect framework that is utilized to maintain standards globally, referred to as the OSI reference model. The OSI reference model is codified and formally defines the concept of network architectures in a layered format. At each stage of data transmission, the OSI reference model uses defined operationally descriptive processes that describe what happens at each stage or layer. The OSI reference model can be referred to as the seven layer model or the OSI 'stack' for data communication. OSI stack is the term utilized within this document (Table A.2.1). (Insert OSI stack diagram)

***Table A.2.1: OSI stack layer functions***

| Layer | Name of the Layer | Functions |
|---|---|---|
| 1 | Physical | • Establish and detach connections, define voltage and data rates, convert data bits into electrical signal<br>• Decide whether transmission is simplex, half-duplex, or full duplex |
| 2 | Data link | • Synchronize, detect error, and correct |

| | | |
|---|---|---|
| | | • Wait for acknowledgment for each transmitted frame |
| 3 | Network | • Route essential signals<br>• Divide outgoing message into packets.<br>• Act as network controller for routing data |
| 4 | Transport | • Decide whether transmission should be parallel or single path, and then multiplex, split or segment the data is required<br>• Break data into smaller units for efficient handling |
| 5 | Session | • Manage synchronized conversation between two systems<br>• Control logging on and off, user authentication, billing, and session management |
| 6 | Presentation | • Concerned with the syntax and semantics of the information transmitted<br>• Known as translating layer |
| 7 | Application | • Retransfer files of information<br>• Assist in LOGIN, check password |

## THE TCP/IP REFERENCE MODEL

The Internet came from the original ARPANET research project that was sponsored by the US Department of Defense as a collaboration effort with many universities and US government organizations. The TCP/IP model was developed when satellite and radio networks were added to the original leased telephone network system. This was to fix the model/s in use at that time which broke down with the addition of these different forms of data transmission. The TCP/IP model is a combination of Transmission Control Protocol (TCP) and Internet Protocol (IP). The goals of the TCP/IP model were:

- Seamless connectivity between multiple networks
- Survival of existing communication upon the loss of subnet architecture
- Various applications with divergent requirements should be successfully handled through the use of a flexible architecture

As TCP/IP was developed for military use, it is able to be used by diverse networks through its flexibility and is robust and able to handle failure. TCP/IP is the most popular interconnectivity reference model and is the reference model / protocol that governs the Internet. The TCP/IP reference model is simpler than the OSI stack, and has only 4 levels: Network, Internet, Transport, and Application.

### Network

Using a protocol, the host connects to the network. The Network layer of the TCP/IP stack corresponds to both the Physical and Data Link layer of the OSI stack. The protocol utilized varies between network and hosts.

### Internet

The Internet layer allows a host to inject packets onto the network, ensuring independent travel to its destination by defining the packet protocol and format. The protocol format created is called the Internet Protocol (IP). The delivery of the IP packets to the correct destination is entrusted to the Internet layer through packet routing and congestion control.

### Transport

The Transport layer, which is situated above the Internet layer allows source and destination entries to communicate with each other. Transmission Control Protocol (TCP) and User Datagram Packets (UDP) are the end-to-end protocols defined by this layer. The function of these protocols allows byte stream communication to be transmitted from one machine to another without introducing errors. The Transport layer also establishes flow control of packets between source and destination machines.

### Application

The Application layer contains all the high level protocols. The Session and Presentation layers defined in the OSI stack are of little importance when transmitting packets upon the Internet, and are therefore not part of the TCP/IP reference model.

## APPENDIX A.3 INTERNET AND OSI STACK PROTOCOL EXAMPLES

Examples of protocols and their relative levels in both the OSI and TCP/IP stacks are demonstrated below in Table A.3.1.

***Table A.3.1: Layer specific protocol examples of both TCP/IP and OSI stacks***

| OSI model | TCP/IP model | Protocols |
|---|---|---|
| Application layer | Application layer | BitTorrent, BootP, DNS, DHCP, FTP, HTTP, HTTPS, NTP, SSH, SMTP, SNMP, Telnet, TFTP, XMPP, SIP, RDP, POP3, IMAP |
| Presentation layer<br>Session layer | Not Present | TLS<br>NetBEUI, NetBIOS, NFS, NCP |
| Transport layer | Transport layer | SCTP, TCP, UDP, SSL |
| Network layer | Internet layer | ICMP, IGMP, IP4, IP6, IPX, OSPF |
| Data link layer | Network (or Host to Network) | Ethernet, Token Ring, FDDI, HDLC, IEEE 802.11, IEEE 802.16, VLAN, X.25, ARP, RARP |
| Physical |  | layer IEEE 1394, Etherloop, Bluetooth, 1000BASE-SX, ISDN, DSL, SONET/SDH, RS-449 |

## APPENDIX A.4 THE INTERNET OF THINGS (IOT) COMMUNICATION MODELS

There is no current consensus for an IoT communication model. There has been a development of the representation models over time, moving from a three layer architecture model for the IoT to a five layer model representation. The two layers added to the original three layer model address the challenges of security implementation and aligning with business relevance. The following provides a review of some current IoT communication models.

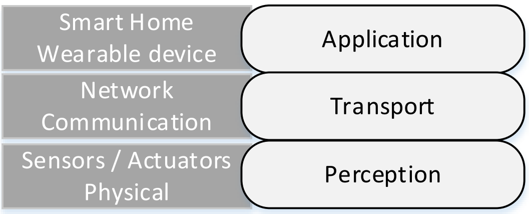


***Figure A.4.1: Three layer IoT Architecture model***

Figure A.4.1 shows the most basic representation of the IoT architecture. The three layers are perception, network and application. The perception layer maps to the physical domain incorporating sensors, transducers and actuators. The role of the perception layer is to present an interface with the world and the environment surrounding the sensor. The perception layer gathers data, which is in turn transmitted via the transport layer. The information from the perception layer is communicated either wirelessly, such as ZigBee, or utilizes a wired transmission format, such as CANbus. The transport layer is also responsible for connecting all the physical entities together and facilitates information sharing between the connected entities. The application layer, in the three layer model, defines all applications that utilize IoT technology. The applications are reliant upon the information gathered from the sensors via the perception layer, therefore the application layer is responsible for providing data services to the IoT applications.

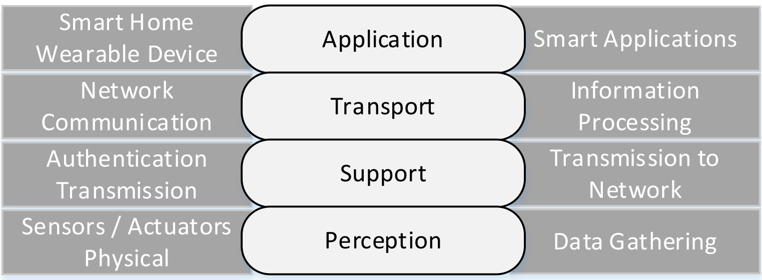


***Figure A.4.2: Four layer IoT Architecture model***

Figure A.4.2 shows an additional layer added to the three layer IoT architecture representation, the support layer. The support layer is designed to accept data from sensors, as output from the perception layer, and provide security services. The data is authenticated, encrypted and transmitted to the network layer via the support layer. Authentication can be through keys, passwords, Pre-shared keys or unique ID settings. The support layer can also be used to pre-condition the data transmission, and depending on the transport medium, can address the data, ready for secure transmission to select application layer smart devices. Transmission control advantages are therefore also supplied by the support layer.

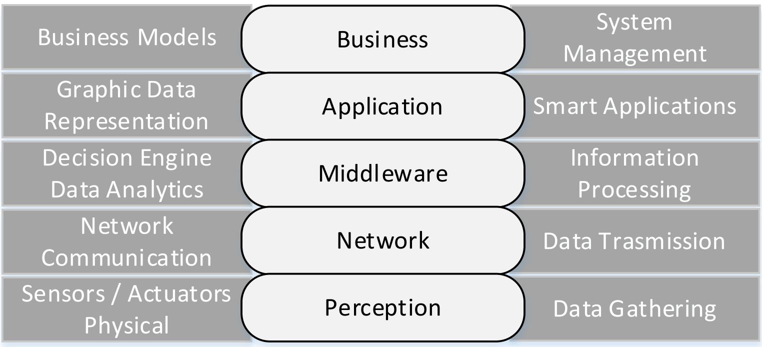


***Figure A.4.3: 5 layer IoT architecture***

The five layer IoT architecture representation model presented as Figure A.4.3, shows the removal of the support layer, and additional layers of middleware and business layers. Gathering data from the sensors is the responsibility of the perception layer. The information is communicated via the network layer via wired and wireless medium to the middleware layer. The middleware layer is responsible for storing, analysing and processing the data gathered from the perception layer, transmitted by the network layer. The middleware layer may utilise database, cloud, or big data processing technologies. The goal of the iddleware layer is to add autonomous decision making and process control. The business layer provides management services to the IoT system. The business layer presents strategic control and functionality directives to system administrators. The business layer utilises graphic data representations and visualisations to inform the administrator at the business layer. The business layer therefore also manages how information is being delivered to the consumer or business and assists with decision making processes.

## APPENDIX B EXPERT SYSTEM DEFINED

This research uses a Design Science approach to produce and test a Prototype for an expert system to assist business managers assess IoT device and services risk. The concept of an Expert System is long established in the literature but the methods and the technologies applied have changed. What an Expert System was in the 1970s and 80s is seen differently today. Today the emphasis is on business value and the former emphasis on computer logic, rules and algorithms has been replaced by business logics. Today the business functionality is paramount and the computing (or artificial intelligence (AI)) element is often customized (rather than built to solve a particular business problem.

Most business software vendors (eg. SAP etc) integrate expert system abilities into their suite of products as general logic solutions. The rule engines are no longer only define the rules an expert would use but are structured to be for any type of complex, volatile, or critical business problem. These Expert Systems are for business process automation and use in challenging new environments (such as Cloud, IoT, and so on). An expert system has two subsystems: the inference engine and the knowledge base. The knowledge base represents facts and rules, and the inference engine applies the rules to the known facts to deduce new facts. In this research inference engine is a Natural Language construct that processes and archives databases of legal and technical documents. The methods for achieving this are specified in Chapter three.

An Expert System is a computer system that emulates the decision-making ability of a human expert, and hence provides greater productivity through shared expertise in businesses. Not every business can employ all the expertise required to obtain the highest performance but the access to generalized and cheaper expertise helps. Expert systems are designed to solve complex problems by reasoning through bodies of knowledge, using rule based systems rather than through computer procedural code. The semantic engine adopted and adapted for this research has its own rules based logics and procedures. The researcher had to design the taxonomy creation processes (Chapter two) but the computational intelligence (AI) was provided by the engine. Current Expert Systems absorb new knowledge quickly, and thus update themselves to be current and relevant. Such systems can receive current and updated information as a continuous process of improvement, and design customized expert guidance. All these concerns have been considered and designed into the current protype (see *The Journal of Expert Systems*, Online ISSN: 1468-0394, © John Wiley & Sons Ltd. 37 Volumes for current and previous Expert System design information).

# APPENDIX C CHAPTER 6 DATA AND TRAINING

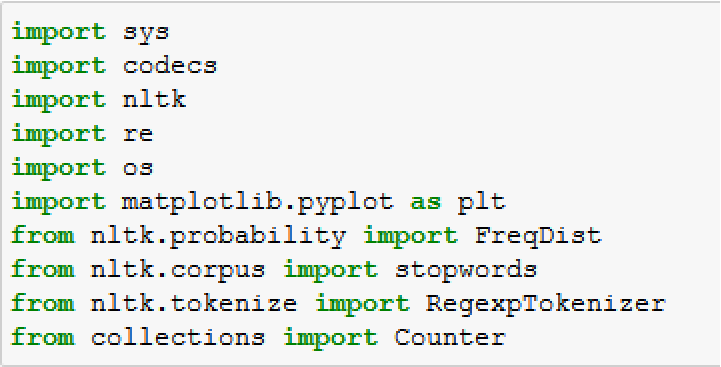
The following sections report all data and processing completed to automate and test the Protype artefact. These are the full findings and a summary is found in Chapter six.

## APPENDIX C.1 MAROOCHY

The following sub-sections report the full data collection, the data processing, the visualization processes, and the algorithm training steps for Maroochy. These data methods are overviewed in Figure 1.1 and are consistent with automation of the expert system manually tested in the Pilot study. The computation of the maturity level values is then completed from these comprehensive outputs and reported in Chapter six.

### Process

In 6.1.1.1

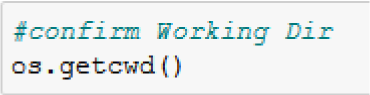


```
import sys
import codecs
import nltk
import re
import os
import matplotlib.pyplot as plt
from nltk.probability import FreqDist
from nltk.corpus import stopwords
from nltk.tokenize import RegexpTokenizer
from collections import Counter
```

Input 6.1.1.1 shows the packages and modules that will be required to perform the various functions within the semantic analysis engine. The **sys** module is imported to provide the Python interpreter information regarding constants, functions and methods, when using the ‘dir(system)’ call. The **codecs** module defines the base classes for standard Python encoders and decoders (codecs). The **codecs** module allow the Python interpreter to encode text to bytes and also encode text to text and bytes to bytes. The error handling lookup process is also managed by the **codecs** module

The **nltk** import command provides access to the Natural Language Tool Kit suite of text processing libraries. The libraries of nltk modules include the tokenization, word frequency and list of default stopwords. The **re** import command provides the Python interpreter regular expression matching operations. Both Unicode strings and 8-bit strings can be searched and matched. Import **os** allows the Python interpreter to interface with the Windows or Linux operating system, enabling cross platform functionality. **matplotlib.pyplot** imports a command style function collection, providing a stateful plotting area in a graphing output, keeping track of current Figures and plotting area. The **matplotlib.pyplot** command has been imported as **plt** to enhance usability. **FreqDist** is imported as a function component of the **nltk.probability** module, allowing the Python interpreter to display the frequency distribution of each word within the selected corpora. Stopwords are imported from the **nltk.corpus** package of texts, providing a default list of **stopwords**, such as ‘and’, ‘if’, ‘then’ and ‘of’.

Input 6.1.1.2



```
#confirm Working Dir
os.getcwd()
```

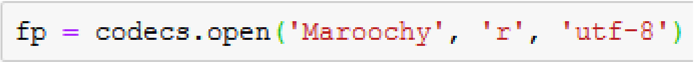
Input 6.1.1.2 Confirms the working directory. Each experiment is constrained to a separate and distinct directory. This assists with corpora delineation, allowing the selection of topic specific corpus.

Output 6.1.1.2

```
'C:\\Users\\User\\Desktop\\PhD\\CH6\\SAE\\Maroochy'
```

Output 6.1.1.2 shows that the working directory is ‘Maroochy’ as part of the Semantic Analysis Engine folder.

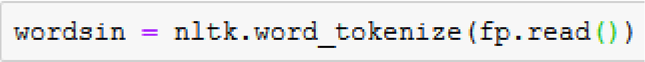
Input 6.1.1.3



```
fp = codecs.open('Maroochy', 'r', 'utf-8')
```

Input 6.1.1.3 queries the working director for the identified corpus, called Maroochy, then open the corpus using the **codecs.open** call. The corpus will then open using the utf-8 formatting structure, and label the output ‘fp’.

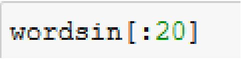
Input 6.1.1.4



```
wordsin = nltk.word_tokenize(fp.read())
```

Input 6.1.1.4 reads the corpus output from ‘Input 6.1.1.3’ labelled ‘fp’, and then tokenizes the results using the **‘nltk.word_tokenize’** call from the **nltk** module. Input 6.1.1.4 provides a tokenized output labelled ‘wordsin’, which word-breaks the entire corpus sentencing structure into single words.

Input 6.1.1.5



```
wordsin[:20]
```

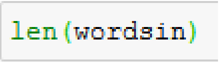
Input6.1.1.5 calls for the first 20 words from the tokenized stream labelled ‘wordsin’ The purpose of Input 6.1.1.5 is to confirm the word-breaking process has been implemented correctly, by retrieving the first 20 words of the Maroochy corpus.

Output 6.1.1.5

```
['Chapter',
 '6',
 'LESSONS',
 'LEARNED',
 'FROM',
 'THE',
 'MAROOCHY',
 'WATER',
 'BREACH',
 'Jill',
 'Slay',
 'and',
 'Michael',
 'Miller',
 'Abstract',
 'Supervisory',
 'control',
 'and',
 'data',
 'acquisition']
```

Output 6.1.1.5 shows the first 20 words of the Maroochy corpus, correctly identifying each separate word, therefore showing that the word-breaking process has occurred.

Input 6.1.1.6



```
len(wordsin)
```

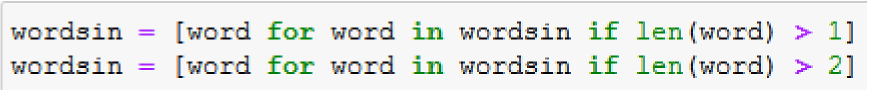
Input 6.1.1.6 calls for a count of the number of words contained within the word-broken output from Input 6.1.1.4 using **len.**

Output 6.1.1.6

```
33222
```

Output 6.1.1.6 provides a primary word count of the entire corpus, with no conditioning or training. The word count output allocates a baseline Figure that will be used to determine the effectiveness of each subsequent training of conditioning sequence.

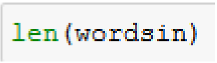
Input 6.1.1.7



```
wordsin = [word for word in wordsin if len(word) > 1]
wordsin = [word for word in wordsin if len(word) > 2]
```

Input 6.1.1.7 is the first of the conditioning sequences. Input 6.1.1.7 parses the ‘wordsin’ file and removes any word token that is less than 2 characters in length, providing an output that overwrites the ‘wordsin’ file. Input 6.1.1.7 also removes any punctuation that has been tokenized.

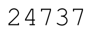
Input 6.1.1.8



```
len(wordsin)
```

Input 6.1.1.8 calls for a count of the number of words contained within the filtered output from the Input 6.1.1.7 process using **len.**

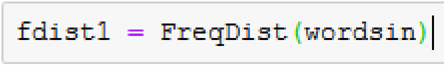
Output 6.1.1.8



```
24737
```

Output 6.1.1.8 shows that 8,485 tokenized words have been removed, with 24,737 tokenized words remaining.

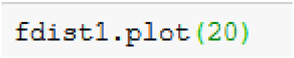
Input 6.1.1.9



```
fdist1 = FreqDist(wordsin)
```

Input 6.1.1.9 calls for an initial frequency distribution of words contained within the Maroochy corpus. The call **FreqDist** provides the frequency distribution of the tokenized words output from 6.1.1.7, and saves the results as ‘fdist1’

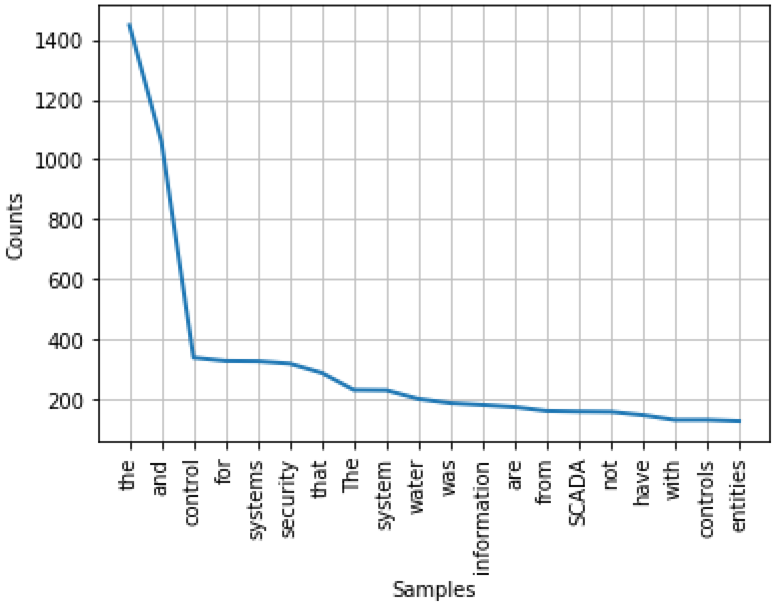
Input 6.1.1.10



```
fdist1.plot(20)
```

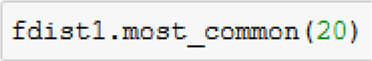
Input 6.1.1.10 calls for a plot graph of the first 20 words within the fdist1 file using **fdist1.plot**

Output 6.1.1.0



Output 6.1.1.10shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Maroochy corpora. The initial plot indicates that there are word tokens that do not add any meaningful input.

Input 6.1.1.11



```
fdist1.most_common(20)
```

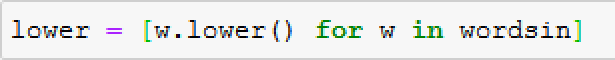
Input 6.1.1.11 calls for a list of the 20 most common words remaining within the Maroochy corpus, identified by frequency, giving the associated word count for each tokenized word.

Output 6.1.1.11

```
('the', 1448),    ***
('and', 1059),    ***
('control', 340),
('for', 329),     ***
('systems', 328),
('security', 320),
('that', 289),    ***
('The', 232),     ***
('system', 231),
('water', 202),
('was', 188),     ***
('information', 182),
('are', 175),     ***
('from', 162),    ***
('SCADA', 160),
('not', 159),     ***
('have', 148),    ***
('with', 132),    ***
('controls', 132),
('entities', 128)
```

Output 6.1.1.11 provides a word count of the most common word occurrences in list format. The output permits identification of words that do not add value to the experiment process. Output 6.1.1.11 indicates that there are 11 words that can be removed from the top 20 occurring words, without removing additional 'meaning' The identified word tokens are indicated with '***'. The out 6.1.1.11 list also shows that words beginning with a capital letter are distinct for words without a capital letter, as seen with the word 'The' and 'the' in the out 6.1.1.11 list.

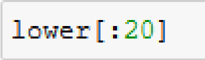
Input 6.1.1.12



```
lower = [w.lower() for w in wordsin]
```

Input 6.1.1.12 parses the tokenized wordlist using the call **lower()** to convert all the word tokens to lowercase. The output is saved as 'lower'

Input 6.1.1.13



```
lower[:20]
```

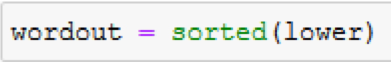
Input 6.1.1.13 calls for a list of the first 20 words in the 'lower' tokenized word stream, to validate the lowercase conversion process of in 6.1.1.12.

Output 6.1.1.13

```
'Chapter',
'lessons',
'learned',
'from',
'the',
'Maroochy',
'water',
'breach',
'jill',
'slay',
'and',
'michael',
'miller',
'abstract',
'supervisory',
'control',
'and',
'data',
'acquisition',
'scada'
```

Output 6.1.1.13 demonstrates that the lower-case conversion process has been effective. The words ‘Chapter’, ‘Maroochy’ and ‘scada’ are all now lower-case.

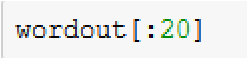
Input 6.1.1.14



```
wordout = sorted(lower)
```

Input 6.1.1.14 sorts the output shown in out 6.1.1.13 into alpha-numeric order, using the call **sorted**, and saves the output as ‘wordout’

Input 6.1.1.15



```
wordout[:20]
```

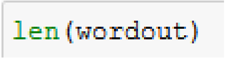
Input 6.1.1.15 calls for a list output of the first 20 entries in the alpha-numerically sorted file labelled wordout.

Output 6.1.1.15

```
'//creativecommons.org/licenses/by-nc-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/sec-cert/ics/index.html',
'//www.courts.qld.gov.au/qjudgment/qca',
'//www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_se
wage/',
'//www.wcc2008.org/site/ifipsampleChapter.pdf',
'//www.whitehouse.gov/omb/memoranda/fy2006/m06-16.pdf',
'//www2.theiet.org/oncomms/sector/computing/library.cfm',
'000',
'000',
'000',
'000',
'000',
'004',
'004',
'004'
```

Output 6.1.1.15 shows that the first 20 entries all little value to the experiment. The output provides a list of conditioning vectors that will be addressed in each of the following steps.

Input 6.1.1.16



```
len(wordout)
```

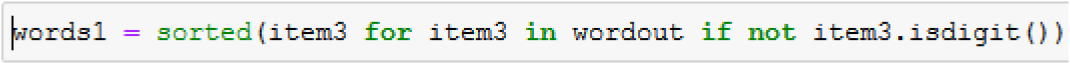
Input 6.1.1.16 calls for the number of tokenized words remaining in the Maroochy corpus. As no words have been removed, the output is compared to out 6.1.1.8 to validate the corpus.

Output 6.1.1.16

```
24737
```

Output 6.1.1.16 is the same as out 6.1.1.8, showing that the conversion to lower-case has not removed any words from the Maroochy corpus.

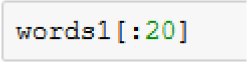
Input 6.1.1.17



```
words1 = sorted(item3 for item3 in wordout if not item3.isdigit())
```

Input 6.1.1.17 parses the ‘wordout’ file produced from 6.1.1.14, identifies and then removes any tokenized words that contain digits using the **isdigit** call. The output is then sorted into alpha-numeric format and then saved as ‘words1’.

Input 6.1.1.18



```
words1[:20]
```

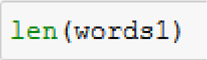
Input 6.1.1.18 calls for a list of the first 20 entries within the ‘words1’ tokenized and sorted

Output 6.1.1.18

```
'//creativecommons.org/licenses/by-nc-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/sec-cert/ics/index.html',
'//www.courts.qld.gov.au/qjudgment/qca',
'//www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_se
wage/'
'//www.wcc2008.org/site/ifipsampleChapter.pdf',
'//www.whitehouse.gov/omb/memoranda/fy2006/m06-16.pdf',
'//www2.theiet.org/oncomms/sector/computing/library.cfm',
'1,157',
'1.8m',
'100,000',
'100,000',
'10:00',
'10:00',
'10:00',
'116,000',
'11pm',
```

Output 6.1.1.18 shows that entries that are only digits have been removed, however any digit string that contains an entry that is not a digit, remains.

Input 6.1.1.19



```
len(words1)
```

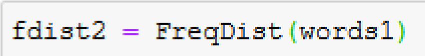
Input 6.1.1.19 calls for a word count of the number of tokenized entire remaining in the Maroochy corpus.

Output 6.1.1.19

```
24548
```

Output 6.1.1.19 shows that 189 tokenized word entries have been removed from the Maroocy corpus.

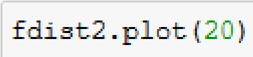
Input 6.1.1.20



```
fdist2 = FreqDist(words1)
```

Input 6.1.1.20 calls for a frequency distribution of words contained within the Maroochy corpus after the first phase of the comprehensive conditioning sequence. The call **FreqDist** provides the frequency distribution of the tokenized words output from 6.1.1.17, and saves the results as ‘fdist2’

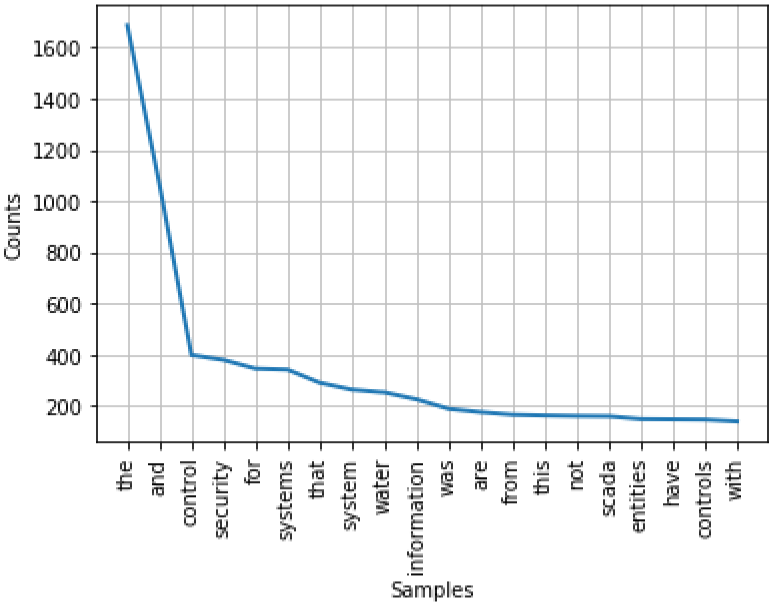
Input 6.1.1.21



```
fdist2.plot(20)
```

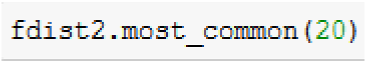
Input 6.1.1.21 calls for a plot graph of the first 20 words within the fdist2 file using **fdist2.plot**

utput6.1.1.21



Output 6.1.1.21 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Maroochy corpus file labelled words1. The initial plot indicates that there are still word tokens remaining that do not add any meaningful input.

Input 6.1.1.22



```
fdist2.most_common(20)
```

Input 6.1.1.22 calls for a list of the 20 most common words remaining within the Maroochy corpus file labelled words1, identified by frequency, providing the associated word count for each tokenized word.

`Output 6.1.1.22

```
('the', 1687),    ***
('and', 1068),    ***
('control', 399),
('security', 380),
('for', 346),     ***
('systems', 342),
('that', 291),    ***
('system', 264),
('water', 253),
('information', 226),
('was', 189),     ***
('are', 176),     ***
('from', 166),    ***
('this', 163),    ***
('not', 161),     ***
('scada', 160),
('entities', 149),
('have', 148),    ***
('controls', 147),
('with', 140)]    ***
```

Output 6.1.1.22 provides a word count of the most common word occurrences in list format. The output permits identification of words that remain after the first stage of conditioning that do not add value to the experimental process. Output 6.1.1.22 indicates that there are 11 words that have not been removed from the top 20 occurring words. These word tokens are indicated with '***'. The out 6.1.1.22 list also shows that words beginning with a capital letter and the words without a capital letter, have been combined, as seen with the word 'the' in the out 6.1.1.22 list

Input 6.1.1.23

```
nltk.download('stopwords')
```

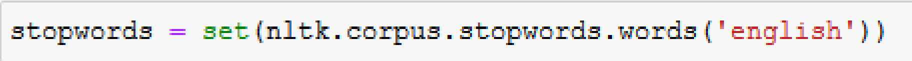
6.1.1.23 downloads the package 'stopwords' from the nltk database in the form of a corpora of stopwords in 16 different languages.

Input 6.1.1.24

```
from nltk.corpus import stopwords
```

Input 6.1.1.24 imports the stopword corpus, ready to be called by the following input 6.1.1.25

Input 6.1.1.25



```
stopwords = set(nltk.corpus.stopwords.words('english'))
```

Input 6.1.1.25 calls for the default stopword corpus containing English stopwords, and sets the output to a file labelled 'stopwords'

Input 6.1.1.26

```
print(stopwords)
```

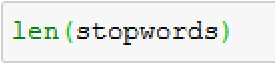
Input 6.1.1.26 calls for a list of the default stopwords contained within stopword file.

Output 6.1.1.26

```
'any', "you're", 'more', 'you', 'had', 'd', 'our', 'hasn',
'myself', "you'd", 'ma', 'haven', 'most', 'has', 'very',
'there', 'mustn', 'is', 'they', 'now', 'to', 'its', 'the',
'off', 'hadn', 'we', 'were', 'again', 'against', 'over', 'didn',
"she's", "isn't", 'not', 'by', 'what', 'so', 'hers', 'needn',
'it', "haven't", 'at', 'as', "mustn't", 'does', 've', 't',
"shan't", 'o', 'too', 'few', "couldn't", 'herself', 'same',
'in', "won't", 'are', 'will', 'shan', 'until', 'my', 's',
'wasn', 'because', 'own', 'she', 'after', 'isn', 'i', 'why',
'down', 'can', 'this', 'wouldn', 'on', 'only', "you'll", 'both',
"wouldn't", 'than', 'a', "weren't", 'who', 'being', 'which',
'such', 'below', "should've", 'weren', "needn't", 'out', 'for',
"doesn't", 'all', 'themselves', 'aren', 'himself', "it's",
'into', 'under', 'ourselves', 'have', "mightn't", 'those',
'itself', 'no', 'whom', 'up', 'll', "hadn't", 'how', "didn't",
'mightn', 'y', 'am', 'do', 'and', 'each', 'here', 'between',
'other', 'm', 'doesn', "you've", 'your', 'ain', 'just',
"that'll", 'doing', 'did', 'these', 'while', 'shouldn', 'me',
'through', 'above', 'their', "wasn't", 'him', 'that',
"shouldn't", "don't", 'of', 'he', 'further', 'if', 'been', 're',
'during', 'where', 'his', 'ours', 'be', 'having', 'her', 'once',
'them', 'then', 'nor', 'yourselves', 'some', 'yourself', 'but',
'with', 'yours', "hasn't", 'couldn', "aren't", 'should', 'was',
'theirs', 'or', 'before', 'from', 'won', 'an', 'don', 'when',
'about'
```

Output 6.1.1.26 displays the list of default English stopwords downloaded from the nltk database.

Input 6.1.1.27



```
len(stopwords)
```

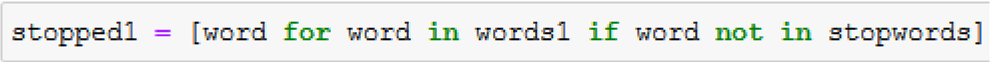
Input 6.1.1.27 calls for a count of the number of stopwords contained within the English stopword corpus.

Output 6.1.1.27

```
179
```

Output 6.1.1.27 shows that there are 179 words contained within the English stopword corpus.

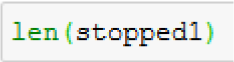
Input 6.1.1.28



```
stopped1 = [word for word in words1 if word not in stopwords]
```

Input 6.1.1.28 parses the words1 tokenized word file, resulting from 6.1.1.17, removing any token contained within the stopword file. The output tokenized file is saved as ‘stopped1’

Input 6.1.1.29



```
len(stopped1)
```

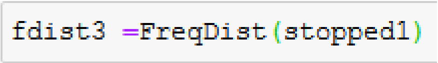
Input 6.1.1.29 calls for a count of the entries contained within the ‘stopped1’ tokenized file, to validate that entries have been removed

Output 6.1.1.29

```
17901
```

Output 6.1.1.29 shows that 6,647 entries have been removed after parsing for any English stopword contained within the default stopword file.

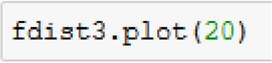
Input 6.1.1.30



```
fdist3 =FreqDist(stopped1)
```

Input 6.1.1.30 calls for a frequency distribution of words contained within the stopped1 tokenized file resulting from the first default stopword conditioning sequence. The call **FreqDist** provides the frequency distribution of the tokenized words output from Input 6.1.1.28, and saves the results as ‘fdist3’

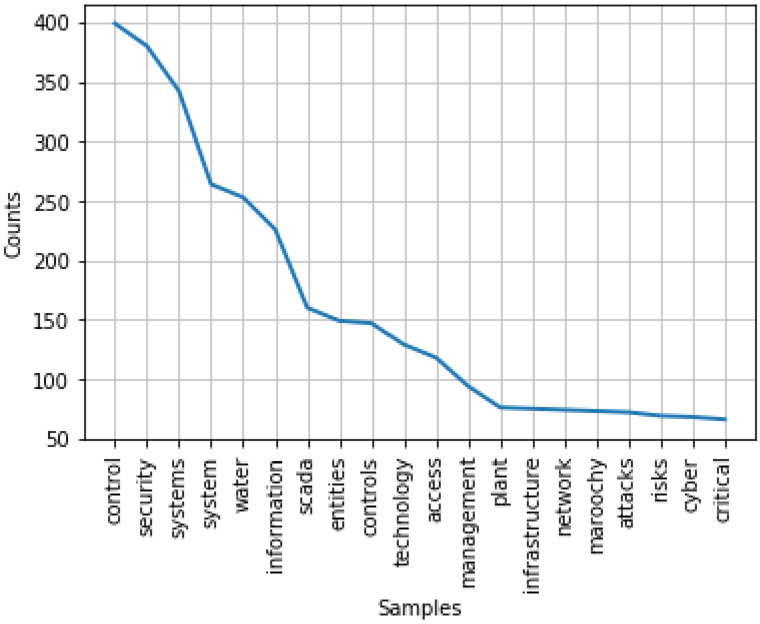
Input 6.1.1.31



```
fdist3.plot(20)
```

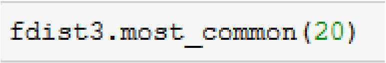
Input 6.1.1.31 calls for a plot graph of the first 20 words within the fdist3 file using **fdist3.plot**

Output 6.1.1.31



Output 6.1.1.31 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Maroochy corpora file, after stopwords have been removed, labelled stopped1. The initial plot indicates that the stop word tokens have been successfully removed.

Input 6.1.1.32



```
fdist3.most_common(20)
```

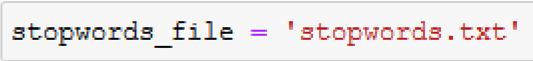
Input 6.1.1.32 calls for a list of the 20 most common words remaining within the Maroochy corpus file labelled stopped1, identified by frequency, providing the associated word count for each tokenized word.

Output 6.1.1.32

```
('control', 399),
('security', 380),
('systems', 342),
('system', 264),
('water', 253),
('information', 226),
('scada', 160),
('entities', 149),
('controls', 147),
('technology', 129),
('access', 118),
('management', 94),
('plant', 76),
('infrastructure', 75),
('network', 74),
('Maroochy', 73),
('attacks', 72),
('risks', 69),
('cyber', 68),
('critical', 66)
```

Output 6.1.1.32 provides a word count of the most common word occurrences contained within the stopped1 file, in list format. The output permits identification of words that remain after the first default stopword removal conditioning, that do not add value to the experiment process. Output 6.1.1.32 indicates that the 11 words that were identified as providing no value, as shown from out 6.1.1.22, are removed from the top 20 occurring words.

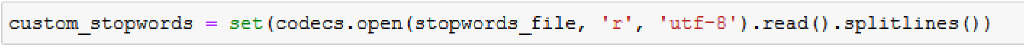
Input 6.1.1.33



```
stopwords_file = 'stopwords.txt'
```

Input 6.1.1.33 links an external text file called ‘stopwords.txt’ that has been created to provide a custom list of domain specific stopwords. The custom stopwords are stored in a text file with a blank first line and each word on a separate line thereafter.

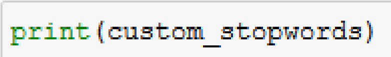
Input 6.1.1.34



```
custom_stopwords = set(codecs.open(stopwords_file, 'r', 'utf-8').read().splitlines())
```

Input 6.1.1.34 parses the text file using the **codecs.open** call, using a **splitlines** parameter that identifies each custom stopword on each new line. The parsed output is then saved as a file labelled custom_stopwords.

Input 6.1.1.35



```
print(custom_stopwords)
```

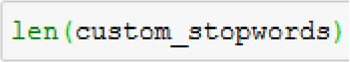
Input 6.1.1.35 calls for a list of the custom stopwords contained within the file ‘custom_stopwords’

Output 6.1.1.35

```
'Table', '7', 'Figure', 'however', 'may', 'nine', 'six', '4',
'eight', 'one', '0', '1', 'fig', 'four', '\ufeff', '9',
'moreover', 'ten', '.', '5', 'would', 'two', '2', '6', 'also',
'five', 'finally', 'Section', '3', '8', 'therefore', 'seven',
'could', 'although', 'three'
```

Output 6.135 provides the list of custom stopwords.

Input 6.1.1.36



```
len(custom_stopwords)
```

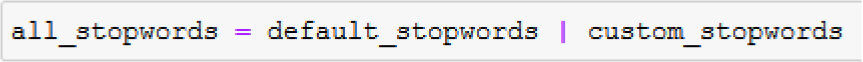
Input 6.1.1.36 calls for a count of the number of entries contained within the 'custom_stopwords' file

Output 6.1.1.36

```
35
```

Output 6.1.1.36 shows that there are 35 entries contained within the custom_stopwords file

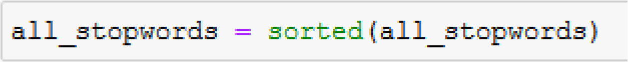
Input 6.1.1.37



```
all_stopwords = default_stopwords | custom_stopwords
```

Input 6.1.1.37 merges the default_stopwords file with the custom_stopwords into a new file labelled all_stopwords

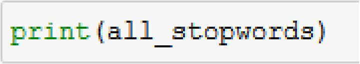
Input 6.1.1.38



```
all_stopwords = sorted(all_stopwords)
```

Input 6.1.1.38 sorts all the entries in the all_stopwords file into alpha-numeric order.

Input 6.1.1.39



```
print(all_stopwords)
```

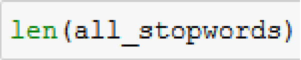
Input 6.1.1.39 calls for a list of all the entries contained within the merged all_stopwords file.

Output 6.1.1.39

```
'.', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a',
'about', 'above', 'after', 'again', 'against', 'ain', 'all',
'also', 'although', 'am', 'an', 'and', 'any', 'are', 'aren',
"aren't", 'as', 'at', 'be', 'because', 'been', 'before',
'being', 'below', 'between', 'both', 'but', 'by', 'can',
'could', 'couldn', "couldn't", 'd', 'did', 'didn', "didn't",
'do', 'does', 'doesn', "doesn't", 'doing', 'don', "don't",
'down', 'during', 'each', 'eight', 'few', 'fig', 'Figure',
'finally', 'five', 'for', 'four', 'from', 'further', 'had',
'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven',
"haven't", 'having', 'he', 'her', 'here', 'hers', 'herself',
'him', 'himself', 'his', 'how', 'however', 'i', 'if', 'in',
'into', 'is', 'isn', "isn't", 'it', "it's", 'its', 'itself',
'just', 'll', 'm', 'ma', 'may', 'me', 'mightn', "mightn't",
'more', 'moreover', 'most', 'mustn', "mustn't", 'my', 'myself',
'needn', "needn't", 'nine', 'no', 'nor', 'not', 'now', 'o',
'of', 'off', 'on', 'once', 'one', 'only', 'or', 'other', 'our',
'ours', 'ourselves', 'out', 'over', 'own', 're', 's', 'same',
'Section', 'seven', 'shan', "shan't", 'she', "she's", 'should',
"should've", 'shouldn', "shouldn't", 'six', 'so', 'some',
'such', 't', 'Table', 'ten', 'than', 'that', "that'll", 'the',
'their', 'theirs', 'them', 'themselves', 'then', 'there',
'therefore', 'these', 'they', 'this', 'those', 'three',
'through', 'to', 'too', 'two', 'under', 'until', 'up', 've',
'very', 'was', 'wasn', "wasn't", 'we', 'were', 'weren',
"weren't", 'what', 'when', 'where', 'which', 'while', 'who',
'whom', 'why', 'will', 'with', 'won', "won't", 'would',
'wouldn', "wouldn't", 'y', 'you', "you'd", "you'll", "you're",
"you've", 'your', 'yours', 'yourself', 'yourselves', '\ufeff'
```

Output 6.1.1.39 provides an alpha-numeric list of the merged all_stopwords file

Input 6.1.1.40



```
len(all_stopwords)
```

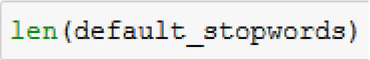
Input 6.1.1.40 calls for a count of all the entries contained within the merged all_stopwords file

Output 6.1.1.40

```
214
```

Output 6.1.1.40 provides a count of 214 entries in the merged all_stopwords file.

Input 6.1.1.41



```
len(default_stopwords)
```

Input 6.1.1.41 calls for a count of all the entries contained within the default_stopwords file

Output 6.1.1.41

```
179
```

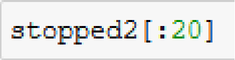
Output 6.1.1.41 provides a count of 179 entries contained within the default_stopwords file, indicating that there have been 35 words added, providing validation confirmation that the custom stop words have been added as required.

Input 6.1.1.42

```
stopped2 = [word for word in stopped1 if word not in all_stopwords]
```

Input 6.1.1.42 parses the stopped1 tokenized word file, resulting from 6.1.1.28, removing any token contained within the custom all_stopwords file. The output tokenized file is saved as ‘stopped2

Input 6.1.1.45



```
stopped2[:20]
```

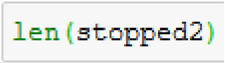
Input 6.1.1.45 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file ‘stopped2’

Output 6.1.1.45

```
'//creativecommons.org/licenses/by-nc-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/sec-cert/ics/index.html',
'//www.courts.qld.gov.au/qjudgment/qca',
'//www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_se
wage/'
'//www.wcc2008.org/site/ifipsampleChapter.pdf',
'//www.whitehouse.gov/omb/memoranda/fy2006/m06-16.pdf',
'//www2.theiet.org/oncomms/sector/computing/library.cfm',
'1,157',
'1.8m',
'100,000',
'100,000',
'10:00',
'10:00',
'10:00',
'116,000',
'11pm',
'120,000'
```

Output 6.1.1.45 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped2’. The list output indicates that there all the first 20 entries identified do not add value to the data, and can therefore be removed from the file.

Input 6.1.1.46



```
len(stopped2)
```

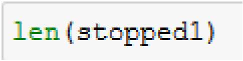
Input 6.1.1.46 calls for a count of the number of entries contained within the ‘stopped2’ file

Output 6.1.1.46

```
17507
```

Output 6.1.1.46 shows that the stopped2 file contains 17,507 tokenized entries

Input 6.1.1.47



```
len(stopped1)
```

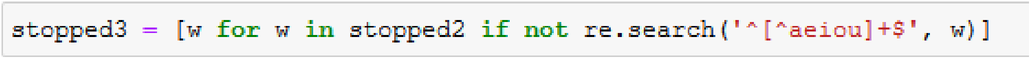
Input 6.1.1.47 calls for a count of the number of entries contained within the ‘stopped1’ file

Output 6.1.1.47

```
17901
```

Output 6.1.1.47 shows that the stopped1 file contains 17,901 tokenized entries, indicating that 394 entries have been removed when applying the custom_stopwords contained within the all_stopwords file.

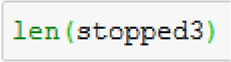
Input 6.1.1.48



```
stopped3 = [w for w in stopped2 if not re.search('^[^aeiou]+$', w)]
```

Input 6.1.1.48 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped2’ file that does not contain a vowel. The output is saved as a tokenized file labelled ‘stopped3’

Input 6.1.1.49



```
len(stopped3)
```

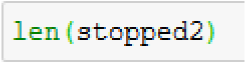
Input 6.1.1.49 calls for a count of the number of entries contained within the ‘stopped3’ file

Output 6.1.1.49

```
17268
```

Output 6.1.1.49 shows that the stopped2 file contains 17,268 tokenized entries contained within the ‘stopped3’ file

Input 6.1.1.50



```
len(stopped2)
```

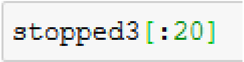
Input 6.1.1.50 calls for a count of the number of entries contained within the ‘stopped2’ file

Output 6.1.1.50

```
17507
```

Output 6.1.1.50 shows that the ‘stopped2’ file contains 17,507 entries, indicating that 239 entries have been removed when applying the regular expression provided by 6.1.1.48 to remove any entry that does not contain a vowel.

Input 6.1.1.51



```
stopped3[:20]
```

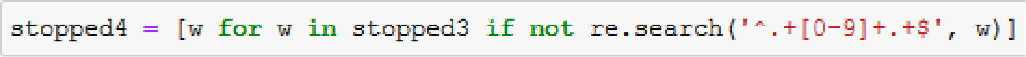
Input 6.1.1.51 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file ‘stopped3’

Output 6.1.1.51

```
//creativecommons.org/licenses/by-nc-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/publications/nistpubs/800-53-',
'//csrc.nist.gov/sec-cert/ics/index.html',
'//www.courts.qld.gov.au/qjudgment/qca',
'//www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_se
wage/'
'//www.wcc2008.org/site/ifipsampleChapter.pdf',
'//www.whitehouse.gov/omb/memoranda/fy2006/m06-16.pdf',
'//www2.theiet.org/oncomms/sector/computing/library.cfm',
'1997-december',
'202002/qca02-164.pdf',
'49-year-old',
'57ea91e4fb429c23',
'9-april',
'abilities',
'ability',
'ability',
'ability',
'ability'
```

Output 6.1.1.51 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped3’. The list output indicates that there are entries that do not add value to the data, and can therefore be removed from the file.

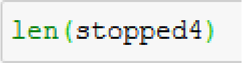
Input 6.1.1.52



```
stopped4 = [w for w in stopped3 if not re.search('^.+[0-9]+.+$', w)]
```

Input 6.1.1.52 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped3’ file that contains a number. The output is saved as a tokenized file labelled ‘stopped4’

Input 6.1.1.53



```
len(stopped4)
```

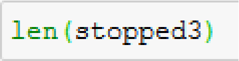
Input 6.1.1.53 calls for a count of the number of entries contained within the ‘stopped4’ file

Output 6.1.1.53

```
17226
```

Output 6.1.1.53 shows that the stopped4 file contains 17,226 tokenized entries

Input 6.1.1.54



```
len(stopped3)
```

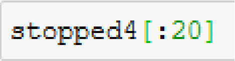
Input 6.1.1.54 calls for a count of the number of entries contained within the ‘stopped3’ file

Output 6.1.1.54

```
17268
```

Output 6.1.1.54 shows that the’ stopped3’ file contains 17,268 tokenized entries, indicating that 42 entries have been removed when applying the regular expression provided by 6.1.1.52 to remove any entry that contains a number.

Input 6.1.1.55



```
stopped4[:20]
```

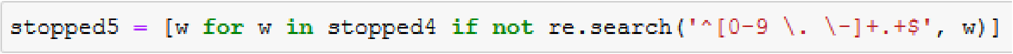
Input 6.1.1.55 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file ‘stopped4’

Output 6.1.1.55

```
'//creativecommons.org/licenses/by-nc-',        ***
'//csrc.nist.gov/sec-cert/ics/index.html',      ***
'//www.courts.qld.gov.au/qjudgment/qca',        ***
'9-april',       ***
'abilities',
'ability',
'ability',
'ability',
'ability',
'ability',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able'
```

Output 6.1.1.55 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped4’. The list output indicates that there is at least one entry that does not add value to the data, and can therefore be removed from the file. The entries identified are highlighted.

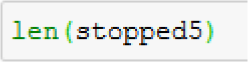
Input 6.1.1.56



```
stopped5 = [w for w in stopped4 if not re.search('^[0-9 \. \-]+.+$', w)]
```

Input 6.1.1.56 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped4’ file that begins with a number. The output is saved as a tokenized file labelled ‘stopped5’

Input 6.1.1.57



```
len(stopped5)
```

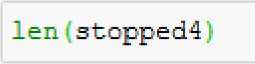
Input 6.1.1.57 calls for a count of the number of entries contained within the ‘stopped5’ file

Output 6.1.1.57

```
17225
```

Output 6.1.1.57 shows that the stopped5 file contains 17,225 tokenized entries

Input 6.1.1.58



```
len(stopped4)
```

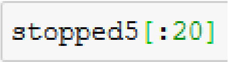
Input 6.1.1.58 calls for a count of the number of entries contained within the 'stopped4' file

Output 6.1.1.58

```
17226
```

Output 6.1.1.58 shows that the 'stopped4' file contains 17,226 entries, indicating that 1 entry have been removed when applying the regular expression provided by 6.1.1.56 to remove any entry that begins with a number.

Input 6.1.1.59



```
stopped5[:20]
```

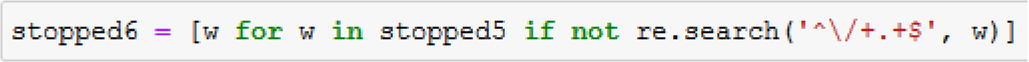
Input 6.1.1.59 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file 'stopped5'

Output 6.1.1.59

```
'//creativecommons.org/licenses/by-nc-',        ***
'//csrc.nist.gov/sec-cert/ics/index.html',      ***
'//www.courts.qld.gov.au/qjudgment/qca',        ***
'abilities',
'ability',
'ability',
'ability',
'ability',
'ability',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able'
```

Output 6.1.1.59 provides a list of the first 20 alpha-numeric tokenized entries contained within the file 'stopped5'. The list output indicates that there are entries that do not add value to the data, and can therefore be removed from the file. The entries identified are highlighted.

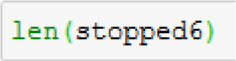
Input 6.1.1.60



```
stopped6 = [w for w in stopped5 if not re.search('^\/+.+$', w)]
```

Input 6.1.1.60 utilizes a regular expression to identify and remove any tokenized entry in the 'stopped5' file that begins with a 'forward slash' as determined by a hyperlink to a URL. The output is saved as a tokenized file labelled 'stopped6'

Input 6.1.1.61



```
len(stopped6)
```

Input 6.1.1.61 calls for a count of the number of entries contained within the ‘stopped6’ file

Output 6.1.1.61

```
17222
```

Output 6.1.1.61 shows that the ‘stopped6’ file contains 17,222 tokenized entries

Input 6.1.1.62

```
len(stopped5)
```

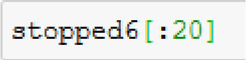
Input 6.1.1.62 calls for a count of the number of entries contained within the ‘stopped5’ file

Output 6.1.1.62

```
17225
```

Output 6.1.1.62 shows that the ‘stopped5’ file contains 17,225 entries, indicating that 3 entries have been removed when applying the regular expression provided by 6.1.1.60 to remove any entry that indicates a URL.

Input 6.1.1.63



```
stopped6[:20]
```

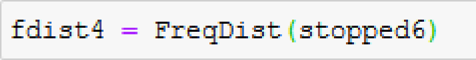
Input 6.1.1.63 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file ‘stopped6’

Output 6.1.1.63

```
'abilities',
'ability',
'ability',
'ability',
'ability',
'ability',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'able',
'abnormal',
'abnormal'
```

Output 6.1.1.63 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped6’. The list output indicates that all entries, within the first 20 identified, add value to the data.

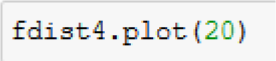
Input 6.1.1.64



```
fdist4 = FreqDist(stopped6)
```

Input 6.1.1.64 calls for a frequency distribution of words contained within the stopped6 tokenized file resulting from the conditioning sequences. The call **FreqDist** provides the frequency distribution of the tokenized words contained within stopped6, and saves the results as ‘fdist4’

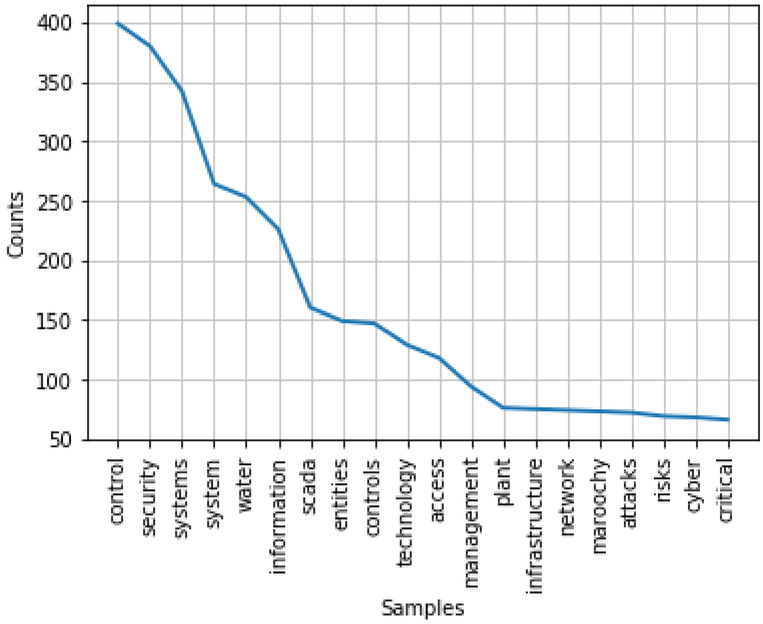
Input 6.1.1.65



```
fdist4.plot(20)
```

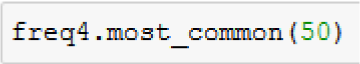
Input 6.1.1.31 calls for a plot graph of the first 20 words within the fdist4 file using **fdist4.plot**

Output 6.1.1.65



Output 6.1.1.65 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Maroochy corpora file, after the conditioning process, labelled fdist4.

Input 6.1.1.66



```
freq4.most_common(50)
```

Input 6.1.1.66 calls for the top 50 entries contained within the ‘stopped6’ tokenized file.

Output 6.1.1.66

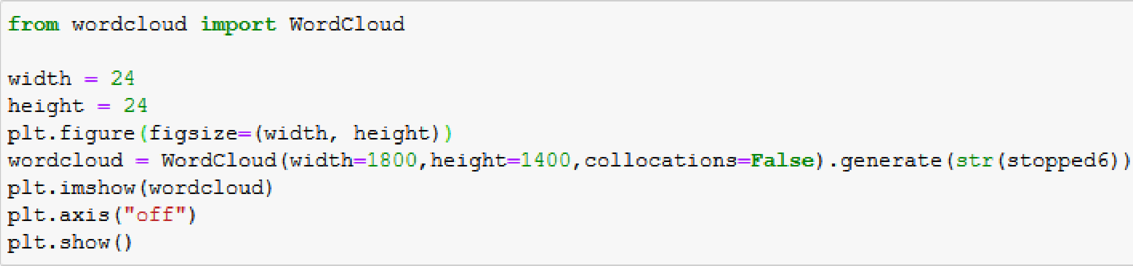
```
('control', 399),              ('audited', 54),
('security', 380),             ('sewage', 53),
('systems', 342),              ('stations', 53),
('system', 264),               ('computer', 51),
('water', 253),                ('risk', 51),
('information', 226),          ('attack', 50),
('scada', 160),                ('use', 50),
('entities', 149),             ('data', 49),
('controls', 147),             ('queensland', 48),
('technology', 129),           ('shire', 47),
('access', 118),               ('australian', 46),
('management', 94),            ('council', 45),
('plant', 76),                 ('services', 45),
('infrastructure', 75),        ('procedures', 44),
('network', 74),               ('process', 44),
('Maroochy', 73),              ('radio', 44),
('attacks', 72),               ('software', 43),
('risks', 69),                 ('identify', 42),
('cyber', 68),                 ('need', 42),
('critical', 66),              ('service', 42),
('audit', 64),                 ('testing', 42),
('station', 64),               ('treatment', 41),
('boden', 63),                 ('providers', 39),
('pumping', 62),               ('used', 39),
('incident', 61),              ('policy', 38)
```

Output 6.1.1.66 shows the output of the 50 most common entries, along with the corresponding word-count value of ‘stopped6’. The file ‘stopped 6’ is the final tokenized output of the conditioning sequences.

**WordCloud Visualization**

The next step is to implement further training and conditioning processes. The output of the training process presents conditioned data, to inform the final risk maturity enumeration. The output from Section 6.1.1 is exhibited in the form of a WordCloud visualization. Using the Word Cloud output as a visualization tool, Output 6.1.2.1 shows that the word ‘vehicle’ has been identified as the word that occurs most frequently. As the word ‘vehicle does not add benefit to the risk identification process, the word is a good example to demonstrate the training abilities of the prototype. The training presents a conditioned output, that allows further reduction of ‘noise’ words, as identified in the Section 6.1.1 process.

Input 6.1.2.1

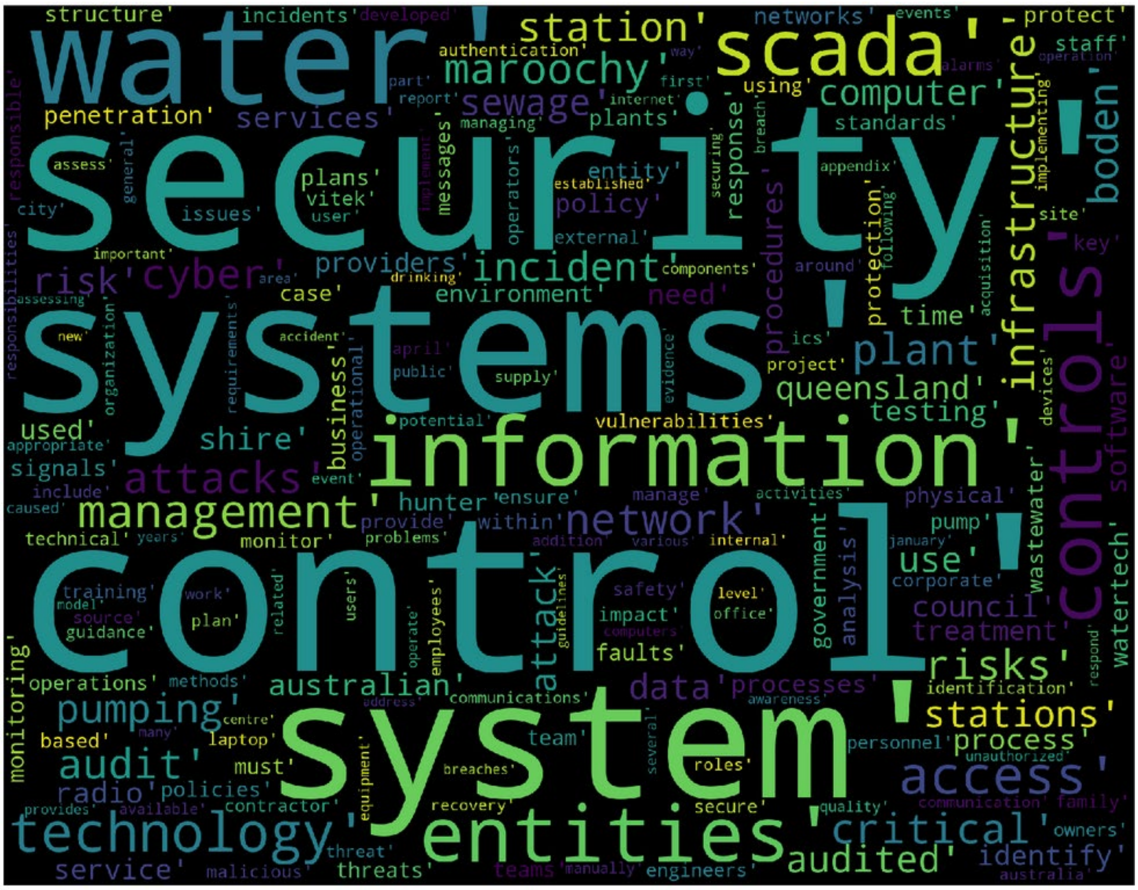


```
from wordcloud import WordCloud

width = 24
height = 24
plt.figure(figsize=(width, height))
wordcloud = WordCloud(width=1800,height=1400,collocations=False).generate(str(stopped6))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Input 6.1.2.1 calls for the WordCloud module and defines the arguments to be implemented. The arguments define parameters of size, position and figure size.

Output 6.1.2.1

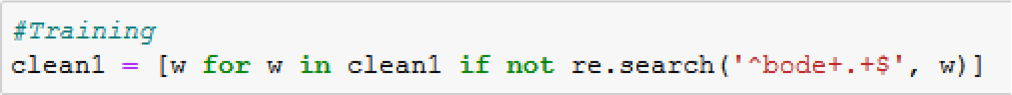


The WordCloud visualization presented as Output 6.1.2.1 shows that the most frequent words ‘control’, ‘security’ and ‘systems’ add value to the risk identification process.

### Training

The output from the WordCloud visualization (Output 6.1.2.1) shows that the words ‘boden’, ‘maroochy’ ‘queensland’ and ‘shire’ are identified as ‘noise’ and can be removed from the word frequency output. The final output presents a trained word frequency list that is conditioned to better inform the final step of maturity modelling. The following steps show the training process.

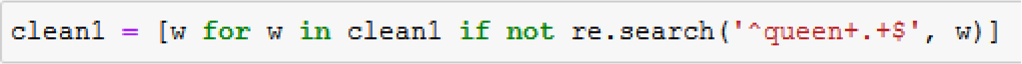
Input 6.1.3.1



```
#Training
clean1 = [w for w in clean1 if not re.search('^bode+.+$', w)]
```

Input 6.1.3.1 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘bode’, which includes the word ‘boden’. The output is saved as a tokenized file labelled ‘clean1’ overwriting the existing file.

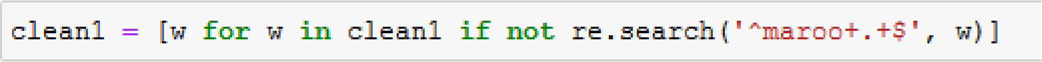
Input 6.1.3.2



```
clean1 = [w for w in clean1 if not re.search('^queen+.+$', w)]
```

Input 6.1.3.2 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘queen’, which includes the word ‘queensland’. The output is saved as a tokenized file labelled ‘clean1’ overwriting the existing file.

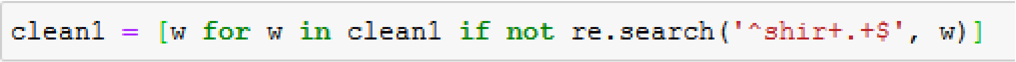
Input 6.1.3.3



```
clean1 = [w for w in clean1 if not re.search('^maroo+.+$', w)]
```

Input 6.1.3.3 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘maroo’, which includes the words ‘maroochy and ‘maroochydore’. The output is saved as a tokenized file labelled ‘clean1’ overwriting the existing file.

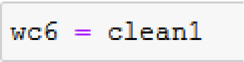
Input 6.1.3.4



```
clean1 = [w for w in clean1 if not re.search('^shir+.+$', w)]
```

Input 6.1.3.4 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘shir’, which includes the word ‘shire’. The output is saved as a tokenized file labelled ‘clean1’ overwriting the existing file.

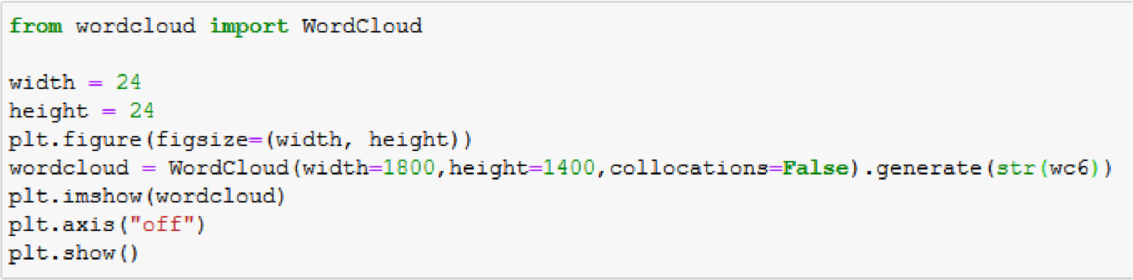
Input 6.1.3.5



```
wc6 = clean1
```

The conditioned tokenized file ‘clean1’ is saved as ‘wc6’ by Input 6.1.3.5 for WordCloud generation.

Input 6.1.3.6

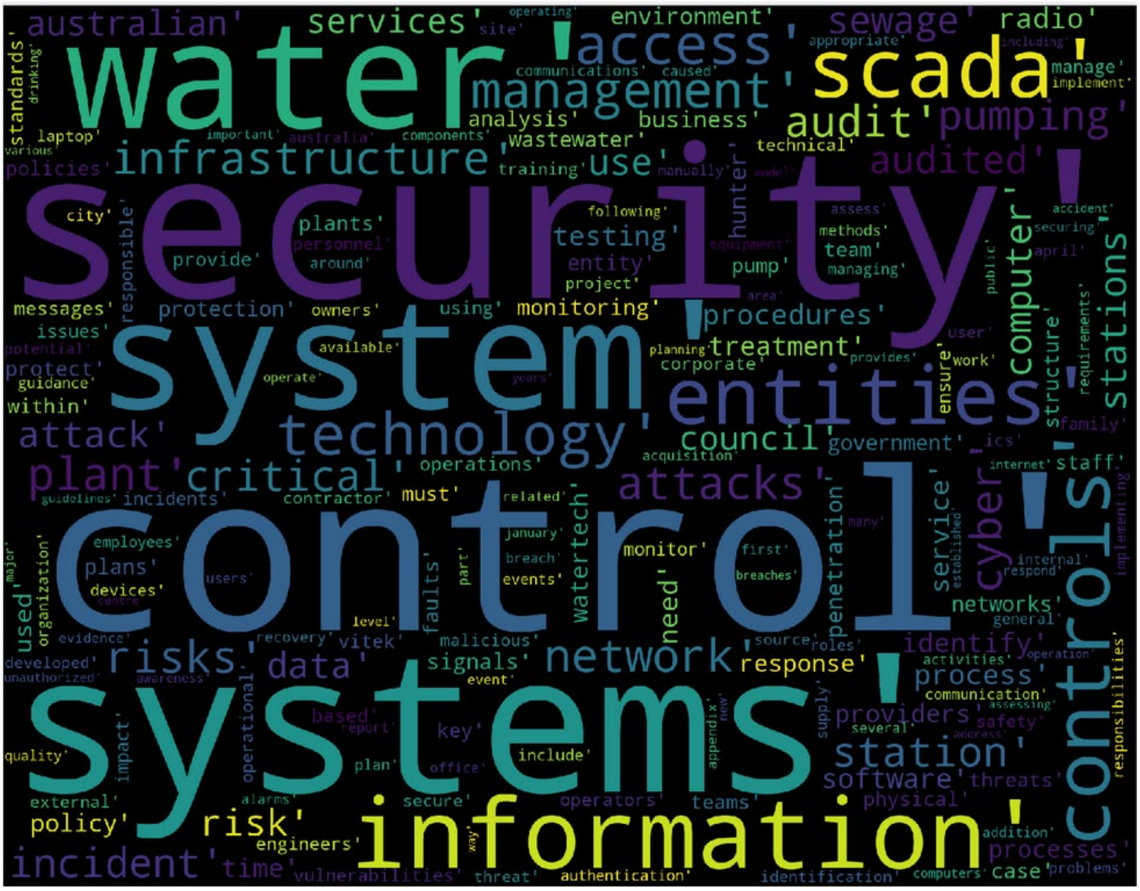


```
from wordcloud import WordCloud

width = 24
height = 24
plt.figure(figsize=(width, height))
wordcloud = WordCloud(width=1800,height=1400,collocations=False).generate(str(wc6))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

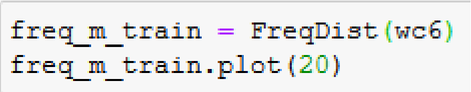
Input 6.1.3.6 resets the WordCloud module and calls for a new visualization using the ‘wc6’ tokenized file. The output is utilized to determine whether the training process has been effective, and presents a conditioned output.

Output 6.1.3.6



The WordCloud visualization presented as Output 6.3.3.2 shows that the words ‘boden’, ‘queensland’, ‘maroochy’ and ‘shire’ are no longer present. The visualization demonstrates the tokenized output is conditioned and provides an effective input for the final maturity analysis process.

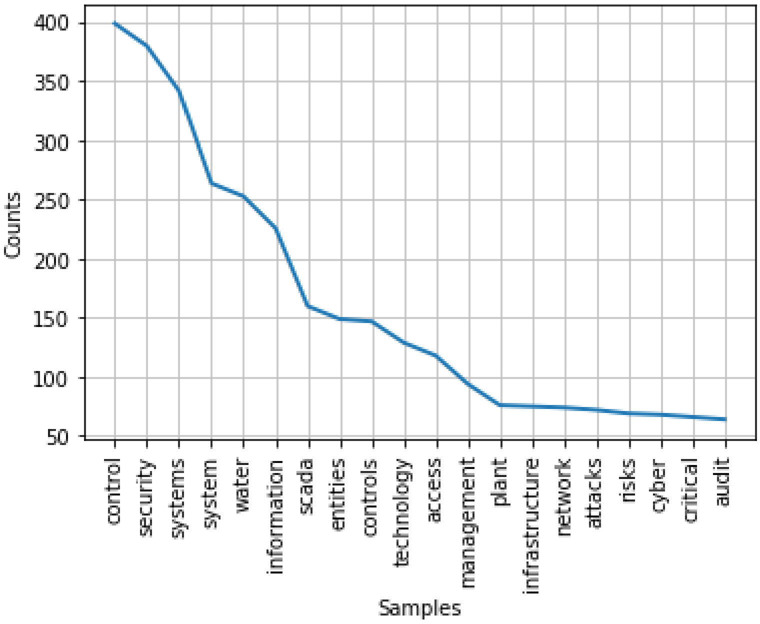
Input 6.1.3.7



```
freq_m_train = FreqDist(wc6)
freq_m_train.plot(20)
```

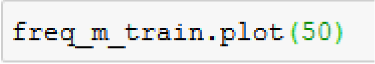
Input 6.1.3.7 calls for a frequency distribution of words contained within the wc6 tokenized file resulting from the training conditioning sequence. The call **FreqDist** provides the frequency distribution of the tokenized words output from Input 6.1.3.5, and saves the results as ‘freq_m_train’ The call **freq_m_train.plot(20)** presents a graph of the frequency count of the first 20 words contained within the conditioned file wc6.

Output 6.1.3.7



Output 6.1.3.7 shows a frequency count plot of the first 20 words, when categorized by frequency, contained within the Maroochy corpora file, after training and, labelled wc6. The plot output indicates that the word token ‘maroochy’ has been successfully removed.

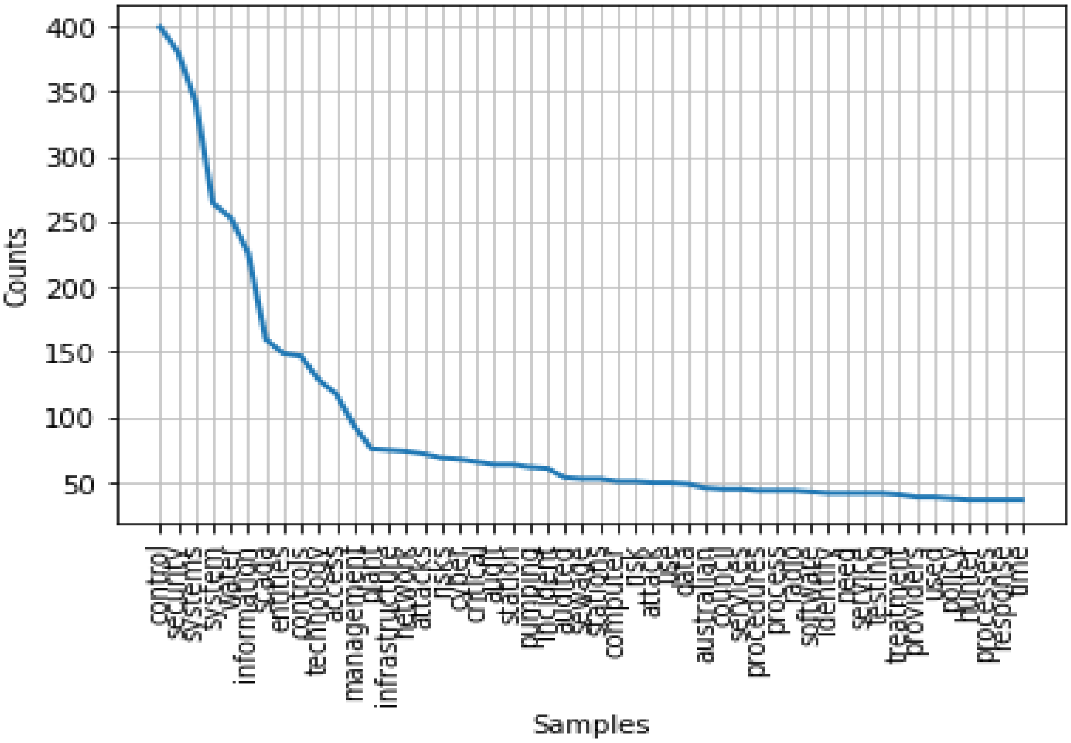
Input 6.1.3.8



```
freq_m_train.plot(50)
```

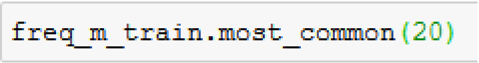
Input 6.3.3.6 uses **freq_m_train.plot(50)** to call for a graph of the frequency count of the first 50 words contained within the conditioned file wc6.

Output 6.1.3.8



Output 6.1.3.8 shows a frequency count plot of the first 50 words, when categorized by frequency, contained within the conditioned Maroochy corpora file, after training. The plot output indicates that the word token ‘boden’, ‘queensland, ‘shire’ and ‘maroochy’ have been successfully removed

Input 6.1.3.9



```
freq_m_train.most_common(20)
```

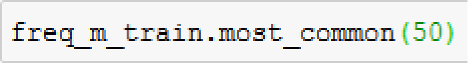
Input 6.3.3.7 calls for a list output of the 20 most common words remaining within the Maroochy corpus conditioned and trained file, identified by frequency, providing the associated word count for each tokenized word.

Output 6.1.3.9

```
('control', 399),
('security', 380),
('systems', 342),
('system', 264),
('water', 253),
('information', 226),
('scada', 160),
('entities', 149),
('controls', 147),
('technology', 129),
('access', 118),
('management', 94),
('plant', 76),
('infrastructure', 75),
('network', 74),
('attacks', 72),
('risks', 69),
('cyber', 68),
('critical', 66),
('audit', 64)]
```

Output 6.1.3.9 presents a word count of the 20 most common word occurrences contained within the Maroochy conditioned and trained file, in list format.

Input 6.1.3.10



```
freq_m_train.most_common(50)
```

Input 6.1.3.10 calls for a list of the 50 most common words remaining within the Maroochy corpus conditioned and trained file, identified by frequency, providing the associated word count for each tokenized word.

Output 6.1.3.10

```
('control', 399),               ('stations', 53),
('security', 380),              ('computer', 51),
('systems', 342),               ('risk', 51),
('system', 264),                ('attack', 50),
('water', 253),                 ('use', 50),
('information', 226),           ('data', 49),
('scada', 160),                 ('australian', 46),
('entities', 149),              ('council', 45),
('controls', 147),              ('services', 45),
('technology', 129),            ('procedures', 44),
('access', 118),                ('process', 44),
('management', 94),             ('radio', 44),
('plant', 76),                  ('software', 43),
('infrastructure', 75),         ('identify', 42),
('network', 74),                ('need', 42),
('attacks', 72),                ('service', 42),
('risks', 69),                  ('testing', 42),
('cyber', 68),                  ('treatment', 41),
('critical', 66),               ('providers', 39),
('audit', 64),                  ('used', 39),
('station', 64),                ('policy', 38),
('pumping', 62),                ('hunter', 37),
('incident', 61),               ('processes', 37),
('audited', 54),                ('response', 37),
('sewage', 53),                 ('time', 37)]
```

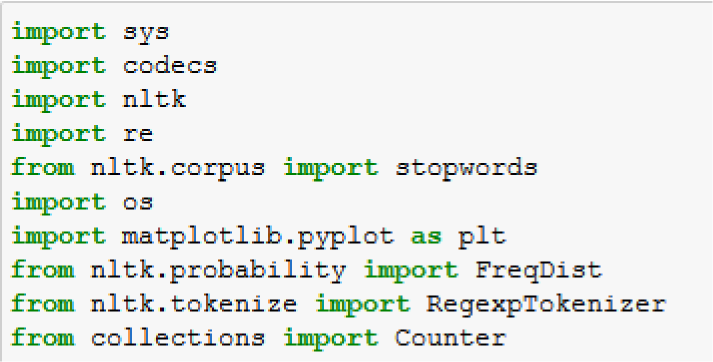
Output 6.3.3.8 shows a list of the 50 most common entries, along with the corresponding word-count value. The output represents the final SAE output to inform the following Maturity Analysis process.

## APPENDIX C.2 TARGET

The following sub-sections report the full data collection, the data processing, the visualization processes, and the algorithm training steps. These data methods are overviewed in Figure 1.1 and are consistent with automation of the expert system manually tested in the Pilot study. The computation of the maturity level values is then completed from these comprehensive outputs and reported in Chapter six.

**Process**

Input 6.2.1.1



```
import sys
import codecs
import nltk
import re
from nltk.corpus import stopwords
import os
import matplotlib.pyplot as plt
from nltk.probability import FreqDist
from nltk.tokenize import RegexpTokenizer
from collections import Counter
```

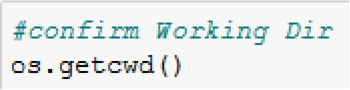
Input 6.2.1.1 shows the packages and modules that will be required to perform the various functions within the semantic analysis engine.

The **sys** module is imported to provide the Python interpreter information regarding constants, functions and methods, when using the ‘dir(system)’ call. The **codecs** module defines the base classes for standard Python encoders and decoders (codecs). The **codecs** module allow the Python interpreter to encode text to bytes and also encode text to text and bytes to bytes. The error handling lookup process is also managed by the **codecs** module. The **nltk** import command provides access to the Natural Language Tool Kit suite of text processing libraries. The libraries of nltk modules include the tokenization, word frequency and list of default stopwords. The **re** import command provides the Python interpreter regular expression matching operations. Both Unicode strings and 8-bit strings can be searched and matched. Import **os** allows the Python interpreter to interface with the Windows or Linux operating system, enabling cross platform functionality.

**matplotlib.pyplot** imports a command style function collection, providing a stateful plotting area in a graphing output, keeping track of current Figures and plotting area. The **matplotlib.pyplot** command has been imported as **plt** to enhance usability. **FreqDist** is imported as a function component of the **nltk.probability** module, allowing the Python interpreter to display the frequency distribution of each word within the selected corpora.

Stopwords are imported from the **nltk.corpus** package of texts, providing a default list of **stopwords**, such as ‘and’, ‘if’, ‘then’ and ‘of’.

Input 6.2.1.2



```
#confirm Working Dir
os.getcwd()
```

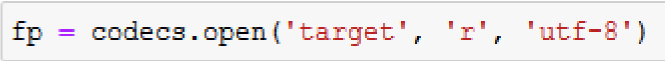
Input 6.2.1.2 Confirms the working directory. Each experiment is constrained to a separate and distinct directory. This assists with corpora delineation, allowing the selection of topic specific corpus.

Output 6.2.1.2

```
'C:\\Users\\User\\Desktop\\PhD\\CH6\\SAE\\Target'
```

Output 6.2.1.2 shows that the working directory is 'Target' as part of the Semantic Analysis Engine folder.

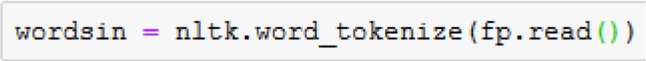
Input 6.2.1.3



```
fp = codecs.open('target', 'r', 'utf-8')
```

Input 6.2.1.3 queries the working director for the identified corpus, called Target, then open the corpus using the **codecs.open** call. The corpus will then open using the utf-8 formatting structure, and then label the output 'fp'.

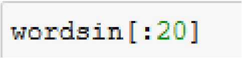
Input 6.2.1.4



```
wordsin = nltk.word_tokenize(fp.read())
```

Input 6.2.1.4 reads the corpus output from 'Input 6.2.1.3' labelled 'fp', and then tokenizes the results using the **'nltk.word_tokenize'** call from the **nltk** module. Input 6.2.1.4 provides a tokenized output labelled 'wordsin', which word-breaks the entire corpus sentencing structure into single words.

Input 6.2.1.5



```
wordsin[:20]
```

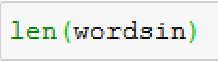
Input6.2.1.5 calls for the first 20 words from the tokenized stream labelled 'wordsin' The purpose of Input 6.2.1.5 is to confirm the word-breaking process has been implemented correctly, by retrieving the first 20 words of the Target corpus.

Output 6.2.1.5

```
'Breaking',
'the',
'Target',
':',
'An',
'Analysis',
'of',
'Target',
'Data',
'Breach',
'and',
'Lessons',
'Learned',
'Xiaokui',
'Shu',
',',
'Ke',
'Tian*',
',',
'Andrew'
```

Output 6.2.1.5 shows the first 20 words of the Target corpus, correctly identifying each separate word, therefore showing that the word-breaking process has occurred.

Input 6.2.1.6



```
len(wordsin)
```

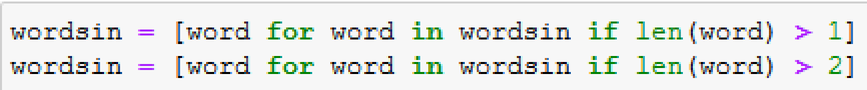
Input 6.2.1.6 calls for a count of the number of words contained within the word-broken output from Input 6.2.1.4 using **len.**

Output 6.2.1.6

```
129331
```

Output 6.2.1.6 provides a primary word count of the entire corpus, with no conditioning or training. The word count output allocates a baseline Figure that will be used to determine the effectiveness of each subsequent training of conditioning sequence.

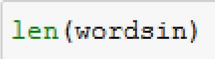
In 6.2.1.7



```
wordsin = [word for word in wordsin if len(word) > 1]
wordsin = [word for word in wordsin if len(word) > 2]
```

Input 6.2.1.7 is the first of the conditioning sequences. Input 6.2.1.7 parses the ‘wordsin’ file and removes any word token that is less than 2 characters in length, providing an output that overwrites the ‘wordsin’ file. Input 6.2.1.7 also removes any punctuation that has been tokenized.

Input 6.2.1.8



```
len(wordsin)
```

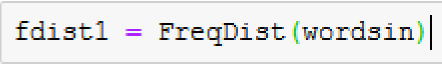
Input 6.2.1.8 calls for a count of the number of words contained within the filtered output from the Input 6.2.1.7 process using **len.**

Output 6.2.1.8

```
92059
```

Output 6.2.1.8 shows that 37,272 tokenized words have been removed, with 92,059 tokenized words remaining.

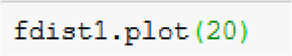
Input 6.2.1.9



```
fdist1 = FreqDist(wordsin)
```

Input 6.2.1.9 calls for an initial frequency distribution of words contained within the Target corpus. The call **FreqDist** provides the frequency distribution of the tokenized words output from 6.2.1.7, and saves the results as ‘fdist1’

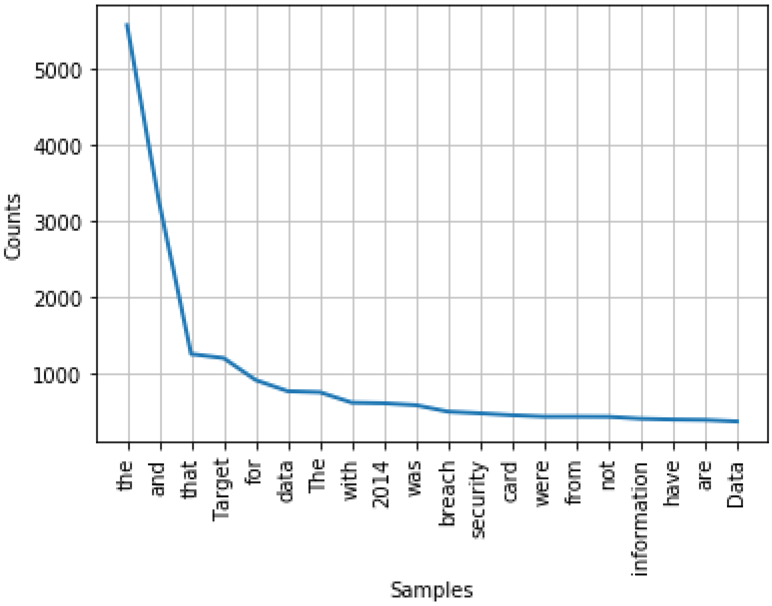
Input 6.2.1.10



```
fdist1.plot(20)
```

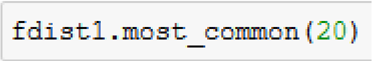
Input 6.2.1.10 calls for a plot graph of the first 20 words within the fdist1 file using **fdist1.plot**

Output 6.2.1.10



Output 6.2.1.10 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Target corpora. The initial plot indicates that there are word tokens that do not add any meaningful input.

Input 6.2.1.11



```
fdist1.most_common(20)
```

Input 6.2.1.11 calls for a list of the 20 most common words remaining within the Target corpus, identified by frequency, giving the associated wordcount for each tokenized word.

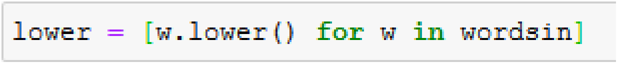
Output 6.2.1.11

```
('the', 5568),    ***
('and', 3243),    ***
('that', 1255),   ***
('Target', 1202),
('for', 913),     ***
('data', 767),
('The', 754),     ***
('with', 617),    ***
('2014', 609),    ***
```

```
('was', 584),     ***
('breach', 499),
('security', 477),
('card', 452),
('were', 434),    ***
('from', 434),    ***
('not', 432),     ***
('information', 406),
('have', 395),    ***
('are', 390),     ***
('Data', 370)
```

Output 6.2.1.11 provides a word count of the most common word occurrences in list format. The output permits identification of words that do not add value to the experiment process. Output 6.2.1.11 indicates that there are 13 words that can be removed from the top 20 occurring words, without removing additional 'meaning' The identified word tokens are indicated by '***'. The out 6.2.1.11 list also shows that words beginning with a capital letter are distinct for words without a capital letter, as seen with the word 'The' and 'the' in the out 6.2.1.11 list.

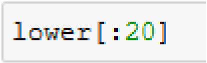
Input 6.2.1.12



```
lower = [w.lower() for w in wordsin]
```

Input 6.2.1.12 parses the tokenized wordlist using the call **lower()** to convert all the word tokens to lowercase. The output is saved as 'lower'

Input 6.2.1.13



```
lower[:20]
```

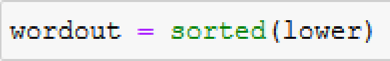
Input 6.2.1.13 calls for a list of the first 20 words in the 'lower' tokenized word stream, to validate the lowercase conversion process of in 6.2.1.12.

Output 6.2.1.13

```
'breaking',
'the',
'target',
'analysis',
'target',
'data',
'breach',
'and',
'lessons',
'learned',
'xiaokui',
'shu',
'tian*',
'andrew',
'ciambrone*',
'and',
'danfeng',
'daphne',
'yao',
'member’
```

Output 6.2.1.13 demonstrates that the lower-case conversion process has been effective. The words ‘target’, and ‘breaking are now lower-case.

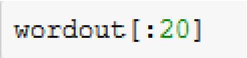
Input 6.2.1.14



```
wordout = sorted(lower)
```

Input 6.2.1.14 sorts the output shown in out 6.2.1.13 into alpha-numeric order, using the call **sorted**, and saves the output as ‘wordout’

Input 6.2.1.15



```
wordout[:20]
```

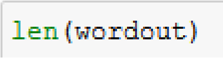
Input 6.2.1.15 calls for a list output of the first 20 entries in the alpha-numerically sorted file labelled wordout.

Output 6.2.1.15

```
'*k.',
'*k.',
',39',
'-15',
'-15',
'-15',
'-30',
'-30',
'-50',
'-50',
'-biggest-data-breaches-hacks/',
'...',
'...',
'...',
'...',
'...',
'...',
'...',
'...',
'...'
```

Output 6.2.1.15 shows that the first 20 entries add no value to the experiment. The output provides a list of conditioning vectors that will be addressed in each of the following steps.

Input 6.2.1.16



```
len(wordout)
```

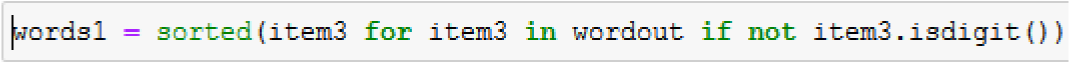
Input 6.2.1.16 calls for the number of tokenized words remaining in the Target corpus. As no words have been removed, the output is compared to out 6.2.1.8 to validate the corpus.

Output 6.2.1.16

```
92059
```

Output 6.2.1.16 is the same as out 6.2.1.8, showing that the conversion to lower-case has not removed any words from the Target corpus.

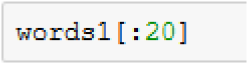
Input 6.2.1.17



```
words1 = sorted(item3 for item3 in wordout if not item3.isdigit())
```

Input 6.2.1.17 parses the ‘wordout’ file produced from 6.2.1.14, identifies and then removes any tokenized words that contain digits using the **isdigit** call. The output is then sorted into alpha-numeric format and then saved as ‘words1’.

Input 6.2.1.18



```
words1[:20]
```

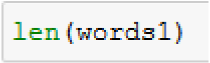
Input 6.2.1.18 calls for a list of the first 20 entries within the ‘words1’ tokenized and sorted

Output

```
'*k.',
'*k.',
',39',
'-15',
'-15',
'-15',
'-30',
'-30',
'-50',
'-50',
'-biggest-data-breaches-hacks/',
'...',
'...',
'...',
'...',
'...',
'...',
'...',
'...',
'...']
```

Output 6.2.1.18 shows that entries that are only digits have been removed, however any digit string that contains an entry that is not a digit, remains.

Input 6.2.1.19



```
len(words1)
```

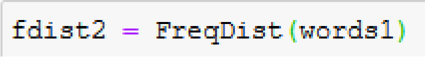
Input 6.2.1.19 calls for a word count of the number of tokenized entire remaining in the Target corpus.

Output 6.2.1.19

```
89675
```

Output 6.2.1.19 shows that 2,384 tokenized word entries have been removed from the Target corpus, when compared to the output 6.2.1.16 of 92,059 tokenized word entries.

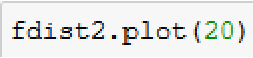
Input 6.2.1.20



```
fdist2 = FreqDist(words1)
```

Input 6.2.1.20 calls for a frequency distribution of words contained within the Target corpus after the first phase of the comprehensive conditioning sequence. The call **FreqDist** provides the frequency distribution of the tokenized words output from 6.2.1.17, and saves the results as ‘fdist2’

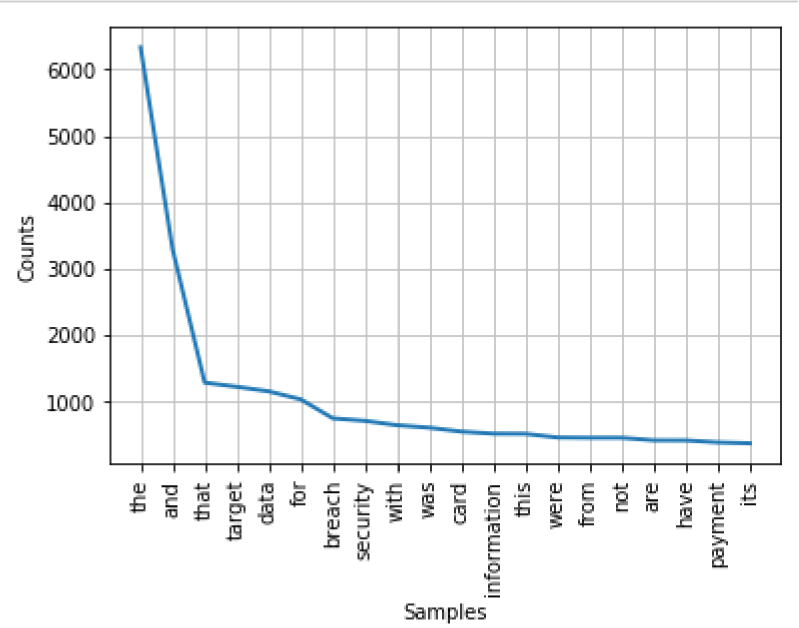
Input 6.2.1.21



```
fdist2.plot(20)
```

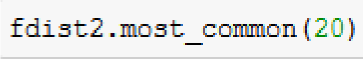
Input 6.2.1.21 calls for a plot graph of the first 20 words within the fdist2 file using **fdist2.plot**

Output 6.2.1.21



Output 6.2.1.21 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Target corpus file labelled words1. The initial plot indicates that there are still word tokens remaining that do not add any meaningful input.

Input 6.2.1.22



```
fdist2.most_common(20)
```

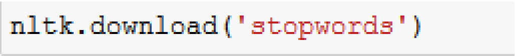
Input 6.2.1.22 calls for a list of the 20 most common words remaining within the Target corpus file labelled words1, identified by frequency, providing the associated word count for each tokenized word.

Output 6.2.1.22

```
('the', 6336),    ***
('and', 3287),    ***
('that', 1275),   ***
('target', 1212),
('data', 1143),
('for', 1022),    ***
('breach', 736),
('security', 698),
('with', 632),    ***
('was', 594),     ***
('card', 535),
('information', 504),
('this', 502),    ***
('were', 446),    ***
('from', 442),    ***
('not', 442),     ***
('are', 404),     ***
('have', 403),    ***
('payment', 373),
('its', 359)]     ***
```

Output 6.2.1.22 provides a word count of the most common word occurrences in list format. The output permits identification of words that remain after the first stage of conditioning that do not add value to the experiment process. Output 6.2.1.22 indicates that there are 13 words that have not been removed from the top 20 occurring words. These word tokens are indicated by '***'. The out 6.2.1.22 list also shows that words beginning with a capital letter and the words without a capital letter, have been combined, as seen with the word 'the' in the out 6.2.1.22 list

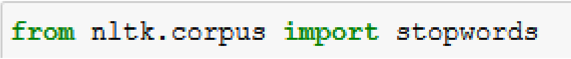
Input 6.2.1.23



```
nltk.download('stopwords')
```

6.2.1.23 downloads the package 'stopwords' from the nltk database. 6.2.1.23 downloads a corpora of stopwords in 16 different languages.

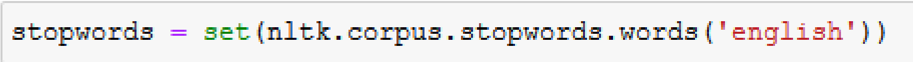
Input 6.2.1.24



```
from nltk.corpus import stopwords
```

Input 6.2.1.24 imports the stopword corpus, ready to be called by the following input 6.2.1.25

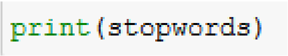
Input 6.2.1.25



```
stopwords = set(nltk.corpus.stopwords.words('english'))
```

Input 6.2.1.25 calls for the default stopword corpus containing English stopwords, and sets the output to a file labelled 'stopwords'

Input 6.2.1.26



```
print(stopwords)
```

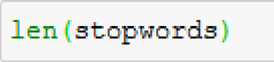
Input 6.2.1.26 calls for a list of the default stopwords contained within stopword file.

Output 6.2.1.26

```
'any', "you're", 'more', 'you', 'had', 'd', 'our', 'hasn',
'myself', "you'd", 'ma', 'haven', 'most', 'has', 'very',
'there', 'mustn', 'is', 'they', 'now', 'to', 'its', 'the',
'off', 'hadn', 'we', 'were', 'again', 'against', 'over', 'didn',
"she's", "isn't", 'not', 'by', 'what', 'so', 'hers', 'needn',
'it', "haven't", 'at', 'as', "mustn't", 'does', 've', 't',
"shan't", 'o', 'too', 'few', "couldn't", 'herself', 'same',
'in', "won't", 'are', 'will', 'shan', 'until', 'my', 's',
'wasn', 'because', 'own', 'she', 'after', 'isn', 'i', 'why',
'down', 'can', 'this', 'wouldn', 'on', 'only', "you'll", 'both',
"wouldn't", 'than', 'a', "weren't", 'who', 'being', 'which',
'such', 'below', "should've", 'weren', "needn't", 'out', 'for',
"doesn't", 'all', 'themselves', 'aren', 'himself', "it's",
'into', 'under', 'ourselves', 'have', "mightn't", 'those',
'itself', 'no', 'whom', 'up', 'll', "hadn't", 'how', "didn't",
'mightn', 'y', 'am', 'do', 'and', 'each', 'here', 'between',
'other', 'm', 'doesn', "you've", 'your', 'ain', 'just',
"that'll", 'doing', 'did', 'these', 'while', 'shouldn', 'me',
'through', 'above', 'their', "wasn't", 'him', 'that',
"shouldn't", "don't", 'of', 'he', 'further', 'if', 'been', 're',
'during', 'where', 'his', 'ours', 'be', 'having', 'her', 'once',
'them', 'then', 'nor', 'yourselves', 'some', 'yourself', 'but',
'with', 'yours', "hasn't", 'couldn', "aren't", 'should', 'was',
'theirs', 'or', 'before', 'from', 'won', 'an', 'don', 'when',
'about'
```

Output 6.2.1.26 displays the list of default English stopwords downloaded from the nltk database.

Input 6.2.1.27



```
len(stopwords)
```

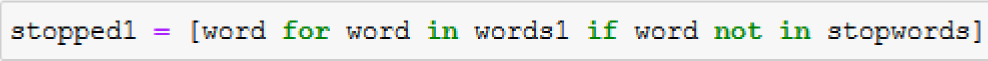
Input 6.2.1.27 calls for a count of the number of stopwords contained within the English stopword corpus.

Output 6.2.1.27

```
179
```

Output 6.2.1.27 shows that there are 179 words contained within the English stopword corpus.

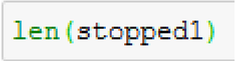
Input 6.2.1.28



```
stopped1 = [word for word in words1 if word not in stopwords]
```

Input 6.2.1.28 parses the words1 tokenized word file, resulting from 6.2.1.17, removing any token contained within the stopword file. The output tokenized file is saved as ‘stopped1’

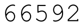
Input 6.2.1.29



```
len(stopped1)
```

Input 6.2.1.29 calls for a count of the entries contained within the ‘stopped1’ tokenized file, to validate that entries have been removed

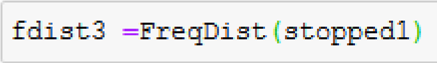
Output 6.2.1.29



```
66592
```

Output 6.2.1.29 shows that 23,083 entries have been removed after parsing for any English stopword contained within the default stopword file, as compared to the output 6.2.1.19, or 89,675 Tokenized word entries

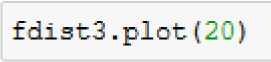
Input 6.2.1.30



```
fdist3 =FreqDist(stopped1)
```

Input 6.2.1.30 calls for a frequency distribution of words contained within the stopped1 tokenized file resulting from the first default stopword conditioning sequence. The call **FreqDist** provides the frequency distribution of the tokenized words output from Input 6.2.1.28, and saves the results as ‘fdist3’

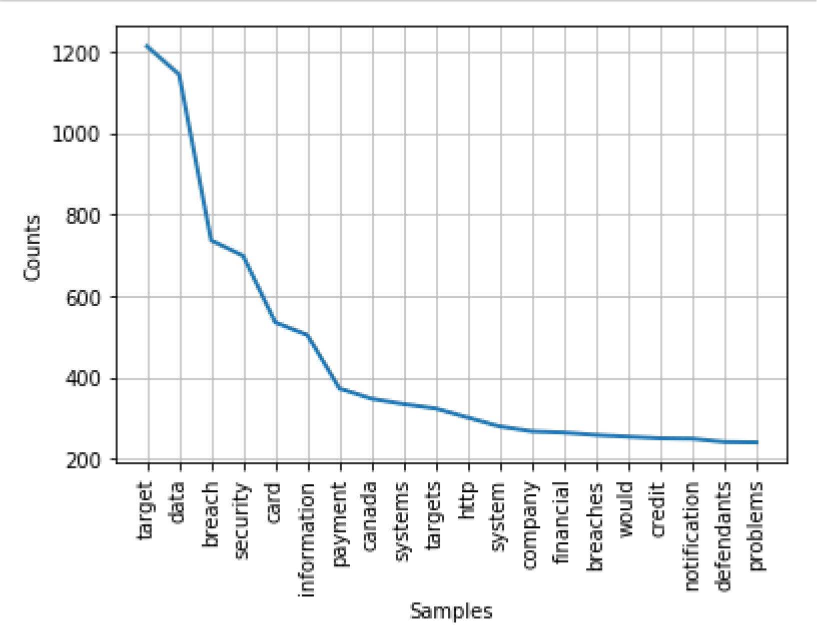
Input 6.2.1.31



```
fdist3.plot(20)
```

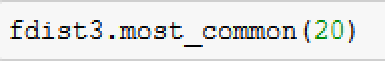
Input 6.2.1.31 calls for a plot graph of the first 20 words within the fdist3 file using **fdist3.plot**

Output 6.2.1.31



Output 6.2.1.31 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Target corpora file, after stopwords have been removed, labelled stopped1. The initial plot indicates that the stop word tokens have been successfully removed.

Input 6.2.1.32



```
fdist3.most_common(20)
```

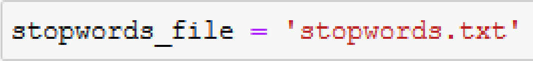
Input 6.2.1.32 calls for a list of the 20 most common words remaining within the Target corpus file labelled stopped1, identified by frequency, providing the associated word count for each tokenized word.

Output 6.2.1.32

```
('target', 1212),
('data', 1143),
('breach', 736),
('security', 698),
('card', 535),
('information', 504),
('payment', 373),
('canada', 348),
('systems', 335),
('targets', 324),
('http', 302),
('system', 280),
('company', 268),
('financial', 265),
('breaches', 259),
('would', 255),          ***
('credit', 251),
('notification', 250),
('defendants', 242),
('problems', 241)
```

Output 6.2.1.32 provides a word count of the most common word occurrences contained within the stopped1 file, in list format. The output permits identification of words that remain after the first default stopword removal conditioning, that do not add value to the experiment process. Output 6.2.1.32 indicates that the 13 words that were identified as providing no value, as shown from out 6.2.1.22, are removed from the top 20 occurring words.

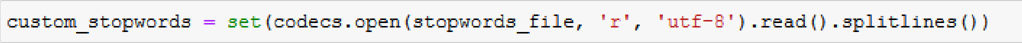
Input 6.2.1.33



```
stopwords_file = 'stopwords.txt'
```

Input 6.2.1.33 links an external text file called ‘stopwords.txt’ that has been created to provide a custom list of domain specific stopwords. The custom stopwords are stored in a text file with a blank first line and each word on a separate line thereafter.

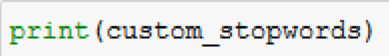
Input 6.2.1.34



```
custom_stopwords = set(codecs.open(stopwords_file, 'r', 'utf-8').read().splitlines())
```

Input 6.2.1.34 parses the text file using the **codecs.open** call, using a **splitlines** parameter that identifies each custom stopword on each new line. The parsed output is then saved as a file labelled custom_stopwords.

Input 6.2.1.35



```
print(custom_stopwords)
```

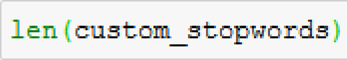
Input 6.2.1.35 calls for a list of the custom stopwords contained within the file 'custom_stopwords'

Output 6.2.1.35

```
'Table', '7', 'Figure', 'however', 'may', 'nine', 'six', '4',
'eight', 'one', '0', '1', 'fig', 'four', '\ufeff', '9',
'moreover', 'ten', '.', '5', 'would', 'two', '2', '6', 'also',
'five', 'finally', 'Section', '3', '8', 'therefore', 'seven',
'could', 'although', 'three'
```

Output 6.135 provides the list of custom stopwords.

Input 6.2.1.36



```
len(custom_stopwords)
```

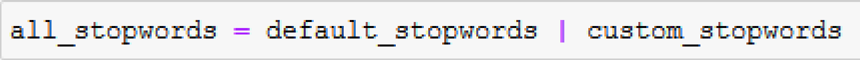
Input 6.2.1.36 calls for a count of the number of entries contained within the 'custom_stopwords' file

Output 6.2.1.36

```
35
```

Output 6.2.1.36 shows that there are 35 entries contained within the custom_stopwords file

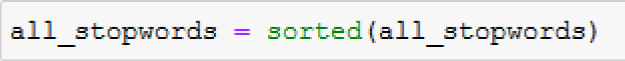
Input 6.2.1.37



```
all_stopwords = default_stopwords | custom_stopwords
```

Input 6.2.1.37 merges the default_stopwords file with the custom_stopwords into a new file labelled all_stopwords

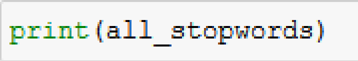
Input 6.2.1.38



```
all_stopwords = sorted(all_stopwords)
```

Input 6.2.1.38 sorts all the entries in the all_stopwords file into alpha-numeric order.

Input 6.2.1.39



```
print(all_stopwords)
```

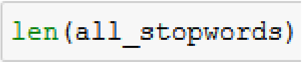
Input 6.2.1.39 calls for a list of all the entries contained within the merged all_stopwords file.

Output 6.2.1.39

```
'.', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a',
'about', 'above', 'after', 'again', 'against', 'ain', 'all',
'also', 'although', 'am', 'an', 'and', 'any', 'are', 'aren',
"aren't", 'as', 'at', 'be', 'because', 'been', 'before',
'being', 'below', 'between', 'both', 'but', 'by', 'can',
'could', 'couldn', "couldn't", 'd', 'did', 'didn', "didn't",
'do', 'does', 'doesn', "doesn't", 'doing', 'don', "don't",
'down', 'during', 'each', 'eight', 'few', 'fig', 'Figure',
'finally', 'five', 'for', 'four', 'from', 'further', 'had',
'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven',
"haven't", 'having', 'he', 'her', 'here', 'hers', 'herself',
'him', 'himself', 'his', 'how', 'however', 'i', 'if', 'in',
'into', 'is', 'isn', "isn't", 'it', "it's", 'its', 'itself',
'just', 'll', 'm', 'ma', 'may', 'me', 'mightn', "mightn't",
'more', 'moreover', 'most', 'mustn', "mustn't", 'my', 'myself',
'needn', "needn't", 'nine', 'no', 'nor', 'not', 'now', 'o',
'of', 'off', 'on', 'once', 'one', 'only', 'or', 'other', 'our',
'ours', 'ourselves', 'out', 'over', 'own', 're', 's', 'same',
'Section', 'seven', 'shan', "shan't", 'she', "she's", 'should',
"should've", 'shouldn', "shouldn't", 'six', 'so', 'some',
'such', 't', 'Table', 'ten', 'than', 'that', "that'll", 'the',
'their', 'theirs', 'them', 'themselves', 'then', 'there',
'therefore', 'these', 'they', 'this', 'those', 'three',
'through', 'to', 'too', 'two', 'under', 'until', 'up', 've',
'very', 'was', 'wasn', "wasn't", 'we', 'were', 'weren',
"weren't", 'what', 'when', 'where', 'which', 'while', 'who',
'whom', 'why', 'will', 'with', 'won', "won't", 'would',
'wouldn', "wouldn't", 'y', 'you', "you'd", "you'll", "you're",
"you've", 'your', 'yours', 'yourself', 'yourselves', '\ufeff'
```

Output 6.2.1.39 provides an alpha-numeric list of the merged all_stopwords file

Input 6.2.1.40



```
len(all_stopwords)
```

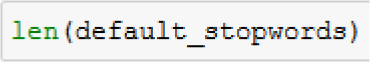
Input 6.2.1.40 calls for a count of all the entries contained within the merged all_stopwords file

Output 6.2.1.40

```
214
```

Output 6.2.1.40 provides a count of 214 entries in the merged all_stopwords file.

Input 6.2.1.41



```
len(default_stopwords)
```

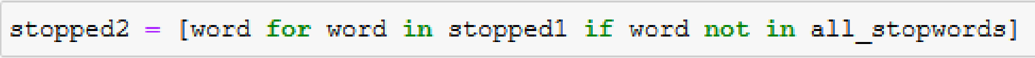
Input 6.2.1.41 calls for a count of all the entries contained within the default_stopwords file

Output 6.2.1.41

```
179
```

Output 6.2.1.41 provides a count of 179 entries contained within the default_stopwords file, indicating that there have been 35 words added, providing validation confirmation that the custom stop words have been added as required.

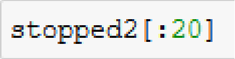
Input 6.2.1.42



```
stopped2 = [word for word in stopped1 if word not in all_stopwords]
```

Input 6.2.1.42 parses the stopped1 tokenized word file, resulting from 6.2.1.28, removing any token contained within the custom all_stopwords file. The output tokenized file is saved as 'stopped2

Input 6.2.1.45



```
stopped2[:20]
```

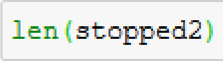
Input 6.2.1.45 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file 'stopped2'

Output 6.2.1.45

```
'*k.',
'*k.',
',39',
'-15',
'-15',
'-15',
'-30',
'-30',
'-50',
'-50',
'-biggest-data-breaches-hacks/',
'...',
'...',
'...',
'...',
'...',
'...',
'...',
'...',
'...'
```

Output 6.1.45 provides a list of the first 20 alpha-numeric tokenized entries contained within the file 'stopped2'. The list output indicates that all the first 20 entries identified do not add value to the data, and can therefore be removed from the file.

Input 6.2.1.46



```
len(stopped2)
```

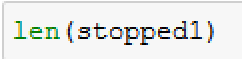
Input 6.2.1.46 calls for a count of the number of entries contained within the 'stopped2' file

Output 6.2.1.46

```
65206
```

Output 6.2.1.46 shows that the stopped2 file contains 65,206 tokenized entries

Input 6.2.1.47



```
len(stopped1)
```

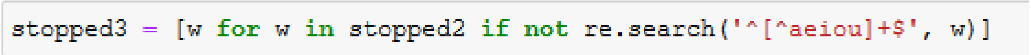
Input 6.2.1.47 calls for a count of the number of entries contained within the ‘stopped1’ file

Output 6.2.1.47

```
66592
```

Output 6.2.1.47 shows that the stopped1 file contains 66,592 tokenized entries, indicating that 1,386 entries have been removed when applying the custom_stopwords contained within the all_stopwords file.

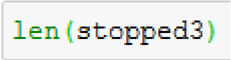
Input 6.2.1.48



```
stopped3 = [w for w in stopped2 if not re.search('^[^aeiou]+$', w)]
```

Input 6.2.1.48 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped2’ file that does not contain a vowel. The output is saved as a tokenized file labelled ‘stopped3’

Input 6.2.1.49



```
len(stopped3)
```

Input 6.2.1.49 calls for a count of the number of entries contained within the ‘stopped3’ file

Output 6.2.1.49

```
63218
```

Output 6.2.1.49 shows that the stopped3 file contains 63,218 tokenized entries contained within the ‘stopped3’ file

Input 6.2.1.50

```
len(stopped2)
```

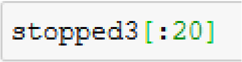
Input 6.2.1.50 calls for a count of the number of entries contained within the ‘stopped2’ file

Output 6.2.1.50

```
65206
```

Output 6.2.1.50 shows that the ‘stopped2’ file contains 65,206 entries, indicating that 1,988 entries have been removed when applying the regular expression provided by 6.2.1.48 to remove any entry that does not contain a vowel.

Input 6.2.1.51



```
stopped3[:20]
```

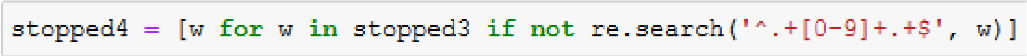
Input 6.2.1.51 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file ‘stopped3’

Output 6.2.1.51

```
'-biggest-data-breaches-hacks/',
'.employed',
'//america.aljazeera.com/multimedia/2014/12/to-catch-a-',
'//america.aljazeera.com/multimedia/2014/12/to-catch-a-',
'//bigstory.ap.org/article/target-data-',
'//bigstory.ap.org/article/target-data-',
'//blog.credit.com/2015/10/target-becomes-',
'//blogs.wsj.com/cio/2014/03/26/retail-association-card-
security-costs-',
'//blogs.wsj.com/cio/2014/03/26/retail-association-card-
security-costs-',
'//blogs.wsj.com/cio/2014/03/26/retail-association-card-
security-costs-outweigh-',
'//blogs.wsj.com/cio/2014/03/26/retail-association-card-
security-costs-outweigh-',
'//blogs.wsj.com/riskandcompliance/2014/02/07/banks-heap-suits-
on-',
'//blogs.wsj.com/riskandcompliance/2014/02/07/banks-heap-suits-
on-',
'//consumerist.com/2014/01/17/non-target-customers-',
'//consumermediallc.files.wordpress.com/2014/01/targetemailgrab.
png',
'//corporate',
'//corporate.target.com/about/awards-',
'//corporate.target.com/about/history/target-through-',
'//corporate.target.com/about/shopping-',
'//corporate.target.com/article/2015/07/cyber-fusion-'
```

Output 6.2.1.51 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped3’. The list output indicates that there are entries that do not add value to the data, and can therefore be removed from the file.

Input 6.2.1.52



```
stopped4 = [w for w in stopped3 if not re.search('^.+[0-9]+.+$', w)]
```

Input 6.2.1.52 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped3’ file that contains a number. The output is saved as a tokenized file labelled ‘stopped4’

Input 6.2.1.53

```
len(stopped4)
```

Input 6.2.1.53 calls for a count of the number of entries contained within the ‘stopped4’ file

Output 6.2.1.53

```
62342
```

Input 6.2.1.54

```
len(stopped3)
```

Input 6.2.1.54 calls for a count of the number of entries contained within the ‘stopped3’ file

Output 6.2.1.54

```
63218
```

Output 6.2.1.54 shows that the’ stopped3’ file contains 63,218 tokenized entries, indicating that 876 entries have been removed when applying the regular expression provided by 6.2.1.52 to remove any entry that contains a number.

Input 6.2.1.55

```
stopped4[:20]
```

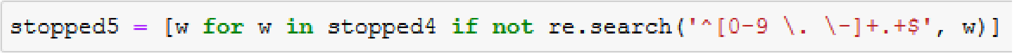
Input 6.2.1.55 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file ‘stopped4’

Output 6.2.1.55

```
'-biggest-data-breaches-hacks/',
'.employed',
'//bigstory.ap.org/article/target-data-',
'//bigstory.ap.org/article/target-data-',
'//corporate',
'//corporate.target.com/about/awards-',
'//corporate.target.com/about/history/target-through-',
'//corporate.target.com/about/shopping-',
'//databreachcalculator.com/calculator/default.aspx',
'//databreachcalculator.com/calculator/default.aspx',
'//datalossdb.org/',
'//edgar.sec.gov/archives/',
'//edgar.sec.gov/archives/edgar/',
'//energycommerce.house.gov/hearing/',
'//energycommerce.house.gov/hearing/',
'//energycommerce.house.gov/hearing/protecting-consumer-
information-',
'//energycommerce.house.gov/hearing/protecting-consumer-
information-',
'//energycommerce.house.gov/sites/',
'//energycommerce.house.gov/sites/',
'//faziomechanical.com/about-us.html'
```

Output 6.2.1.55 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped4’. The list output indicates that there many entries that do not add value to the data, and can therefore be removed from the file. The entries identified are predominantly URL links.

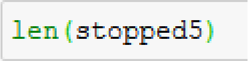
Input 6.2.1.56



```
stopped5 = [w for w in stopped4 if not re.search('^[0-9 \. \-]+.+$', w)]
```

Input 6.2.1.56 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped4’ file that begins with a number. The output is saved as a tokenized file labelled ‘stopped5’

Input 6.2.1.57



```
len(stopped5)
```

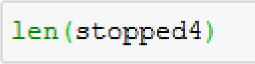
Input 6.2.1.57 calls for a count of the number of entries contained within the ‘stopped5’ file

Output 6.2.1.57

```
62335
```

Output 6.2.1.57 shows that the stopped5 file contains 17,225 tokenized entries

Input 6.2.1.58



```
len(stopped4)
```

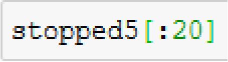
Input 6.2.1.58 calls for a count of the number of entries contained within the ‘stopped4’ file

Output 6.2.1.58

```
62342
```

Output 6.2.1.58 shows that the ‘stopped4’ file contains 62,342 entries, indicating that 7 entries have been removed when applying the regular expression provided by 6.2.1.56 to remove any entry that begins with a number.

Input 6.2.1.59



```
stopped5[:20]
```

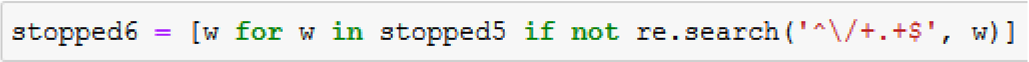
Input 6.2.1.59 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file ‘stopped5’

Output 6.2.1.59

```
'//bigstory.ap.org/article/target-data-',
'//bigstory.ap.org/article/target-data-',
'//corporate',
        '//corporate.target.com/about/awards-',
        '//corporate.target.com/about/history/target-through-',
        '//corporate.target.com/about/shopping-',
        '//databreachcalculator.com/calculator/default.aspx',
        '//databreachcalculator.com/calculator/default.aspx',
        '//datalossdb.org/',
        '//edgar.sec.gov/archives/',
        '//edgar.sec.gov/archives/edgar/',
        '//energycommerce.house.gov/hearing/',
        '//energycommerce.house.gov/hearing/',
        '//energycommerce.house.gov/hearing/protecting-consumer-
        information-',
        '//energycommerce.house.gov/hearing/protecting-consumer-
        information-',
        '//energycommerce.house.gov/sites/',
        '//energycommerce.house.gov/sites/',
        '//faziomechanical.com/about-us.html',
        '//faziomechanical.com/target-',
'//faziomechanical.com/target-breach-statement.pdf'
```

Output 6.2.1.59 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped5’. The list output indicates that there are entries that do not add value to the data, and can therefore be removed from the file. The entries identified are seen to be URL entries.

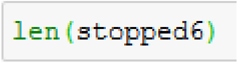
Input 6.2.1.60



```
stopped6 = [w for w in stopped5 if not re.search('^\/+.+$', w)]
```

Input 6.2.1.60 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped5’ file that begins with a ‘forward slash’ as determined by a hyperlink to a URL. The output is saved as a tokenized file labelled ‘stopped6’

Input 6.2.1.61



```
len(stopped6)
```

Input 6.2.1.61 calls for a count of the number of entries contained within the ‘stopped6’ file

Output 6.2.1.61

```
62186
```

Output 6.2.1.61 shows that the ‘stopped6’ file contains 17,222 tokenized entries

Input 6.2.1.62

```
len(stopped5)
```

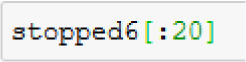
Input 6.2.1.62 calls for a count of the number of entries contained within the ‘stopped5’ file

Output 6.2.1.62

```
62335
```

Output 6.2.1.62 shows that the ‘stopped5’ file contains 32,335 entries, indicating that 149 entries have been removed when applying the regular expression provided by 6.2.1.60 to remove any entry that indicates a URL.

Input 6.2.1.63



```
stopped6[:20]
```

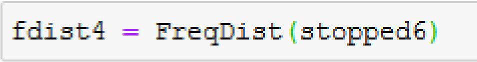
Input 6.2.1.63 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file ‘stopped6’

Output 6.2.1.63

```
'aaa',
'aaa',
'aaron',
'aaron',
'abbreviated',
'abbreviated',
'abilities',
'abilities',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability'
```

Output 6.2.1.63 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped6’. The list output indicates that all entries, within the first 20 identified, add value to the data.

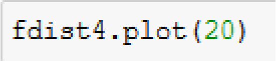
Input 6.2.1.64



```
fdist4 = FreqDist(stopped6)
```

Input 6.2.1.64 calls for a frequency distribution of words contained within the stopped6 tokenized file resulting from the conditioning sequences. The call **FreqDist** provides the frequency distribution of the tokenized words contained within stopped6, and saves the results as ‘fdist4’

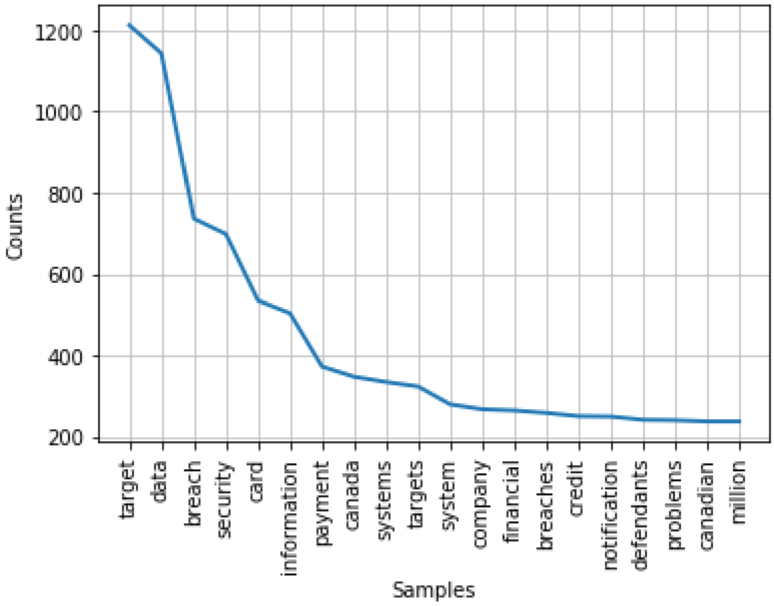
Input 6.2.1.65



```
fdist4.plot(20)
```

Input 6.2.1.31 calls for a plot graph of the first 20 words within the fdist4 file using **fdist4.plot**

Output 6.2.1.65



Output 6.2.1.65 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Target corpora file, after the conditioning process, labelled fdist4.

Input 6.2.1.66

```
freq4.most_common(50)
```

Input 6.2.1.66 calls for the top 50 entries contained within the 'stopped6' tokenized file.

Output 6.2.1.66

```
('target', 1212),          ('act', 185),
('data', 1143),            ('malware', 185),
('breach', 736),           ('business', 175),
('security', 698),         ('law', 168),
('card', 535),             ('cybersecurity', 164),
('information', 504),      ('network', 164),
('payment', 373),          ('personal', 157),
('canada', 348),           ('report', 149),
('systems', 335),          ('companies', 145),
('targets', 324),          ('customers', 145),
('system', 280),           ('states', 145),
('company', 268),          ('congress', 144),
('financial', 265),        ('new', 142),
('breaches', 259),         ('used', 141),
('credit', 251),           ('technology', 138),
('notification', 250),     ('class', 135),
('defendants', 242),       ('state', 135),
('problems', 241),         ('costs', 134),
('canadian', 238),         ('made', 133),
('million', 238),          ('first', 131),
('chain', 236),            ('canadas', 130),
('stores', 231),           ('inventory', 127),
('cards', 210),            ('see', 127),
```

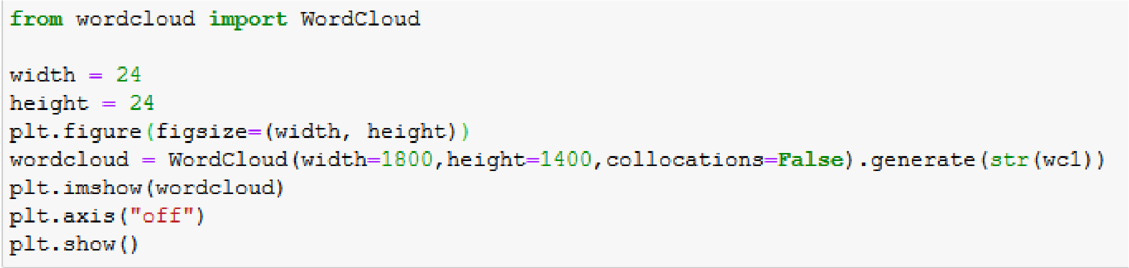
| | |
|---|---|
| ('supply', 208),<br>('federal', 206), | ('pos', 125),<br>('issues', 121)] |

Output 6.2.1.66 shows the output of the 50 most common entries, along with the corresponding word-count value of 'stopped6'. The file 'stopped 6' is the final tokenized output of the stopword conditioning sequences.

**WordCloud Visualisation**

The next step is to implement further training and conditioning processing. The output of the training process presents conditioned data, to inform the final risk maturity enumeration. The output from Section 6.2.1 is exhibited in the form of a WordCloud visualization. Using the Word Cloud output as a visualisation tool, Output 6.2.2.1 shows that the word 'vehicle' has been identified as the word that occurs most frequently. As the word 'vehicle does not add benefit to the risk identification process, the word is a good example to demonstrate the training abilities of the prototype. The training presents a conditioned output that allows further reduction of 'noise' words, as identified in the Section 6.2.1 process.

Input 6.2.2.1

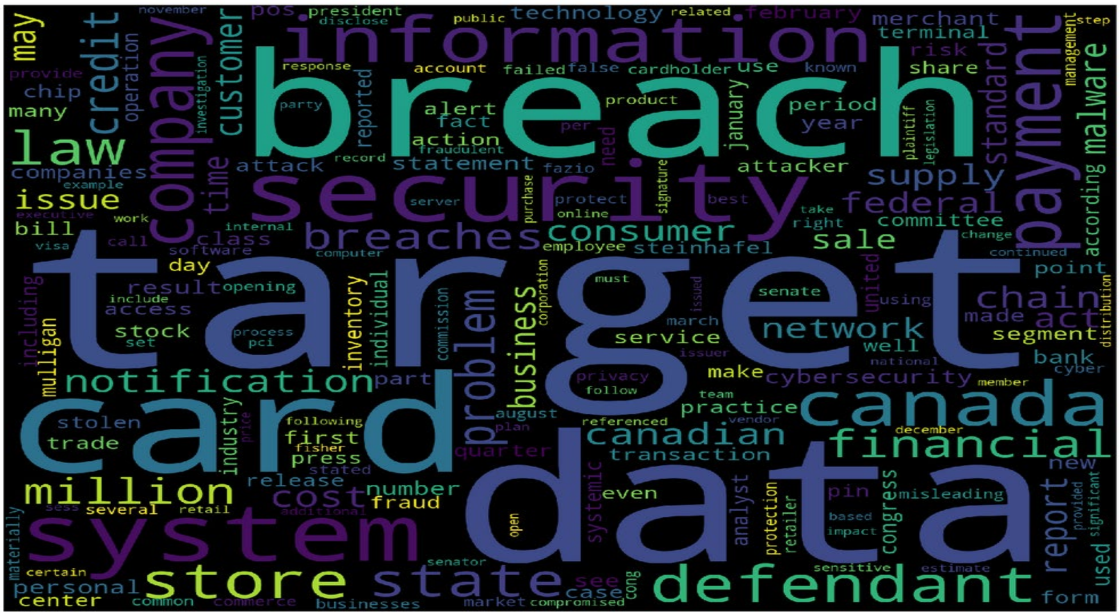


```
from wordcloud import WordCloud

width = 24
height = 24
plt.figure(figsize=(width, height))
wordcloud = WordCloud(width=1800,height=1400,collocations=False).generate(str(wc1))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Input 6.2.2.1 calls for the WordCloud module and defines the arguments to be implemented. The arguments define parameters of size, position and figuresize.

Output 6.2.2.1

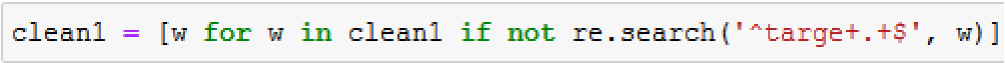


The WordCloud visualization presented as Output 6.3.2.1 shows that the word ‘target’ is identified as the word that occurs most frequently.

### Training

Using the Word Cloud output as a visualization tool, Output 6.2.2.5 shows that the word ‘target’ has been identified as the word that occurs most frequently. As the word ‘target’ does not add benefit to the risk identification process, the word is a good example to demonstrate the training abilities of the prototype. The word ‘breach’ is also selected to demonstrate the conditioning process, as the word does not add benefit to the risk identification process. The training presents a conditioned output that allows further reduction of ‘noise’ words, as identified in the Section 6.2.1 process.

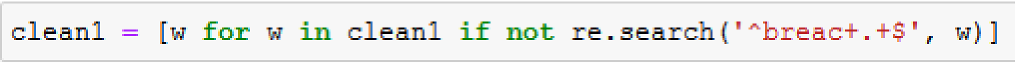
Input 6.2.3.1



```
clean1 = [w for w in clean1 if not re.search('^targe+.+$', w)]
```

Input 6.2.3.1 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘target’, which includes the words ‘target’ and ‘targets’. The output is saved as a tokenized file labeled ‘clean1’, overwriting the existing file.

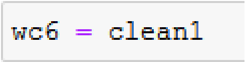
Input 6.2.3.2



```
clean1 = [w for w in clean1 if not re.search('^breac+.+$', w)]
```

Input 6.2.3.2 utilizes a regular expression to identify and remove any tokenized entry in the ‘clean1’ file that contains the word ‘breach’, which includes the words ‘breach’ and ‘breaching’. The output is saved as a tokenized file labeled ‘clean1’, overwriting the existing file

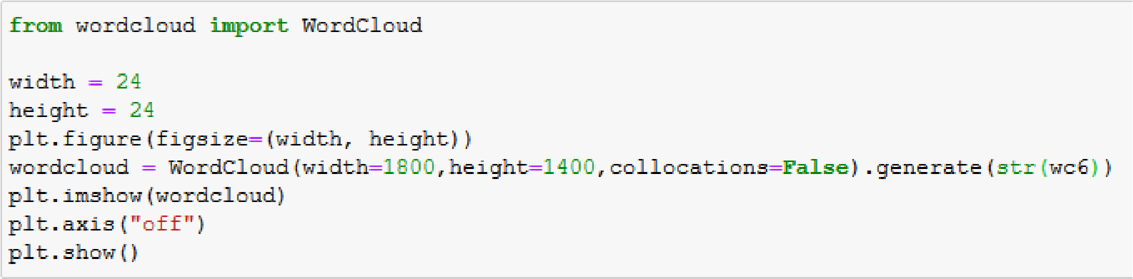
Input 6.2.3.3



```
wc6 = clean1
```

Input 6.2.3.3 saves the file labeled ‘clean1’ as ‘wc6’

Input 6.2.3.4

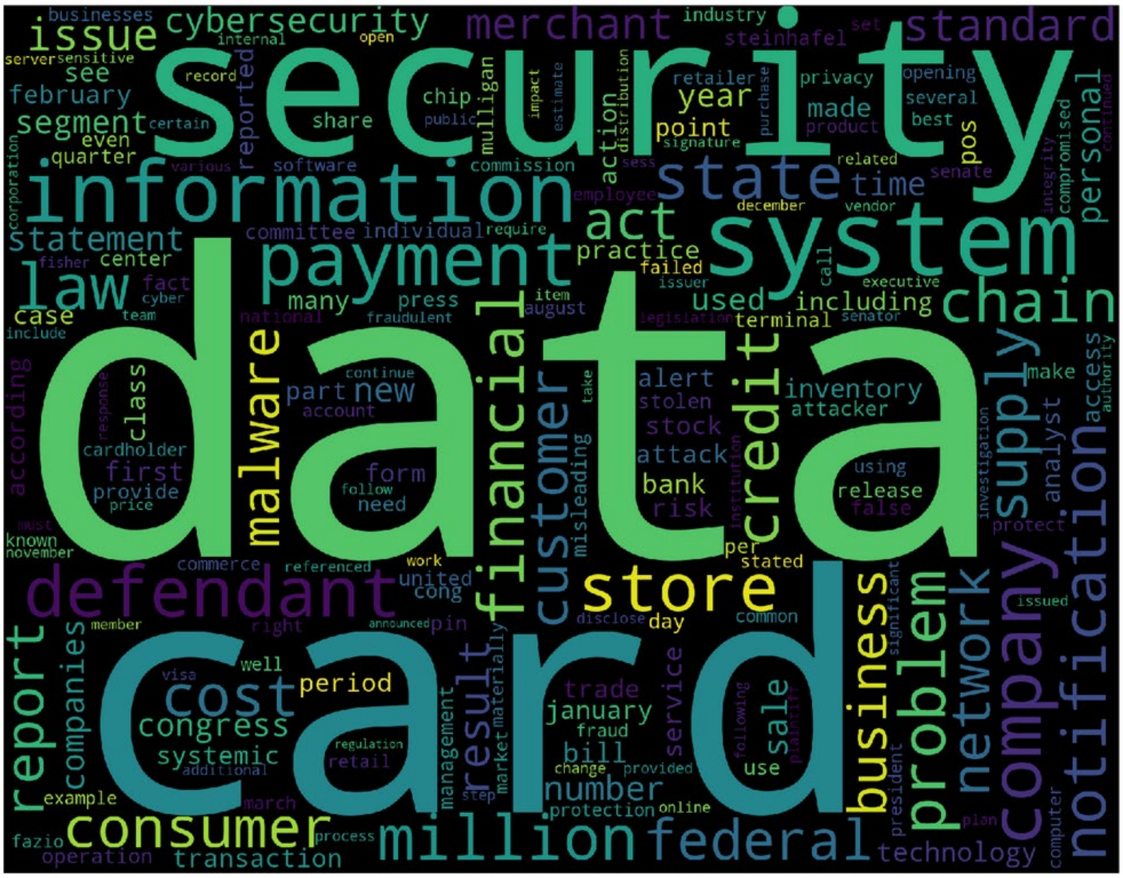


```
from wordcloud import WordCloud

width = 24
height = 24
plt.figure(figsize=(width, height))
wordcloud = WordCloud(width=1800,height=1400,collocations=False).generate(str(wc6))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

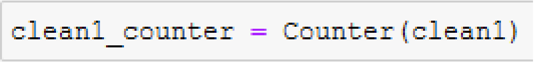
Input 6.2.3.4 resets the WordCloud module and calls for a new visualization using the ‘wc6’ tokenized file. The output is utilized to determine whether the training process has been effective, and presents a conditioned output.

Output 6.2.3.4



The WordCloud visualization presented as Output 6.2.3.3 shows that the word ‘target’ is no longer present. The visualization demonstrates the tokenized output is conditioned and provides an effective input for the final maturity analysis process.

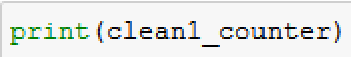
Input 6.2.3.5



```
clean1_counter = Counter(clean1)
```

The file ‘clean1’ is processed with a counter algorithm that counts the frequency occurrence of each word within the trained and conditioned file, and the output is labelled clean1_counter.

Input 6.2.3.6



```
print(clean1_counter)
```

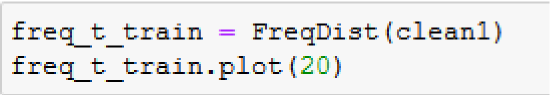
Input 6.2.3.5 calls for a list of the words contained within the conditioned file ‘clean1_counter’ with the corresponding occurrence frequency count.

Output 6.2.3.6

```
'data': 1143, 'security': 698, 'card': 535, 'information': 504,
'payment': 373, 'systems': 335, 'system': 280, 'company': 268,
'financial': 265, 'credit': 251, 'notification': 250, 'defendants':
242, 'problems': 241, 'million': 238, 'chain': 236, 'stores': 231,
'cards': 210, 'supply': 208, 'federal': 206, 'act': 185, 'malware':
185, 'business': 175, 'law': 168, 'cybersecurity': 164, 'network':
164, 'personal': 157, 'report': 149, 'companies': 145, 'customers':
145, 'states': 145, 'congress': 144, 'new': 142, 'used': 141,
'technology': 138, 'class': 135, 'state': 135, 'costs': 134, 'made':
133, 'first': 131, 'inventory': 127, 'see': 127, 'pos': 125, 'issues':
121, 'segment': 120, 'laws': 119, 'consumers': 117, 'including': 117,
'systemic': 117, 'february': 116, 'access': 115, 'trade': 110,
'companys': 109, 'many': 109, 'sales': 109, 'time': 109, 'according':
108, 'form': 108, 'january': 107, 'defendant': 104, 'case': 103,
'period': 102, 'reported': 102, 'results': 101, 'standards': 101,
'stock': 101, 'united': 100, 'consumer': 99, 'practices': 98, 'chip':
97, 'part': 97, 'stolen': 97, 'cost': 95, 'operations': 94, 'press':
94, 'even': 92, 'number': 90, 'statements': 90, 'merchant': 89,
'store': 88, 'committee': 86, 'misleading': 85, 'use': 85, 'quarter':
84, 'release': 84, 'risk': 84, 'services': 84, 'mulligan': 81, 'per':
81, 'false': 80, 'management': 80, 'year': 80, 'action': 79, 'point':
79, 'bill': 78, 'common': 78, 'fraud': 78, 'industry': 78, 'known':
78, 'merchants': 78, 'president': 78, 'privacy': 78, 'march': 77,
'stated': 77, 'failed': 76, 'several': 76, 'standard': 76, 'day': 75,
'commerce': 74, 'protection': 74, 'businesses': 73, 'materially': 73,
'numbers': 73, 'using': 72, 'best': 71, 'call': 71, 'software': 71,
'attackers': 70, 'august': 70, 'compromised': 70, 'national': 70,
'retail': 70, 'senate': 70, 'statement': 70, 'steinhafel': 70,
'protect': 69, 'banks': 68, 'fazio': 67, 'pin': 67, 'well': 67,
'analyst': 66, 'bank': 66, 'commission': 66, 'example': 66, 'set': 66,
'attack': 65, 'continued': 65, 'public': 65, 'result': 65, 'alerts':
64, 'make': 64, 'years': 64, 'december': 63, 'disclose': 63, 'need':
63, 'transaction': 63, 'legislation': 62, 'online': 62, 'related': 62,
'share': 62, 'cyber': 61, 'employees': 61, 'provide': 60, 'bills': 59,
'computer': 59, 'market': 59, 'open': 59, 'following': 58, 'internal':
58, 'opening': 58, 'referenced': 58, 'individual': 57, 'sensitive':
57, 'additional': 56, 'certain': 56, 'fraudulent': 56, 'impact': 56,
'issued': 56, 'provided': 56, 'significant': 56, 'customer': 55,
'distribution': 55, 'fact': 55, 'signature': 55, 'process': 54,
'senator': 54, 'terminals': 54, 'center': 53, 'cong.': 53,
'investigation': 53, 'members': 53, 'november': 53, 'response': 53,
'sess.': 53, 'various': 53, 'cardholder': 52, 'must': 52, 'account':
51, 'reports': 51, 'right': 51, 'sale': 51, 'team': 51, 'visa': 51,
'follows': 50, 'transactions': 50, 'authority': 49, 'executive': 49,
'integrity': 49, 'issue': 49, 'service': 49, 'announced': 48, 'emv':
48, 'fisher': 48, 'losses': 48, 'recklessly': 48, 'address': 47,
'among': 47, 'debit': 47, 'disregarded': 47, 'knew': 47, 'loss': 47,
'whether': 47, 'within': 47, 'based': 46, 'expected': 46, 'herein':
46, 'home': 46, 'much': 46, 'price': 46, 'sec': 46, 'work': 46,
'analysts': 45, 'attacks': 45, 'corporation': 45, 'critical': 45,
'enforcement': 45, 'include': 45, 'likely': 45, 'opened': 45,
'protecting': 45, 'subcommittee': 45, 'across': 44, 'alert': 44,
'changes': 44, 'continue': 44, 'due': 44, 'government': 44, 'like':
44, 'plaintiff': 44, 'required': 44, 'affected': 43, 'exchange': 43,
'key': 43, 'less': 43, 'officer': 43, 'analysis': 42, 'billion': 42,
'different': 42, 'note': 42, 'pci': 42, 'plan': 42, 'potential': 42,
'recent': 42, 'regarding': 42, 'retailers': 42, 'take': 42, 'ability':
41, 'addition': 41, 'better': 41, 'centers': 41, 'ceo': 41,
'conference': 41, 'discussed': 41, 'experienced': 41, 'filed': 41,
'guidance': 41, 'payments': 41, 'private': 41, 'product': 41,
'support': 41, 'acquirer': 40, 'blackpos': 40, 'dilution': 40,
'given': 40, 'hackers': 40, 'infrastructure': 40, 'secure': 40,
'throughout': 40, 'university': 40, 'able': 39, 'another': 39,
'approximately': 39, 'clear': 39, 'general': 39, 'institutions': 39,
'journal': 39, 'material': 39, 'mechanical': 39, 'records': 39,
'times': 39, 'alleged': 38, 'chief': 38, 'current': 38, 'depot': 38,
'eps': 38, 'found': 38, 'get': 38, 'included': 38, 'noted': 38,
'order': 38, 'pertinent': 38, 'sap': 38, 'servers': 38, 'signed': 38,
'testimony': 38, 'allowed': 37, 'called': 37, 'issuers': 37, 'pay':
37, 'since': 37, 'stating': 37, 'still': 37, 'difficult': 36,
'encrypted': 36, 'future': 36, 'investors': 36, 'products': 36,
'second': 36, 'sharing': 36, 'amount': 35, 'effective': 35,
'incident': 35, 'necessary': 35, 'networks': 35, 'securities': 35,
'sent': 35, 'supra': 35, 'addresses': 34, 'detailed': 34, 'earnings':
34, 'john': 34, 'krebs': 34, 'large': 34, 'policy': 34, 'prior': 34,
'reasonable': 34, 'require': 34, 'students': 34, 'total': 34,
'additionally': 33, 'and/or': 33, 'detection': 33, 'email': 33,
'every': 33, 'fireeye': 33, 'forth': 33, 'held': 33, 'important': 33,
'initial': 33, 'media': 33, 'monitoring': 33, 'research': 33,
'terminal': 33, 'already': 32, 'believe': 32, 'directly': 32,
'disclosure': 32, 'insurance': 32, 'internet': 32, 'issuing': 32,
'outside': 32, 'paper': 32, 'performance': 32, 'processors': 32,
'retailer': 32, 'said': 32, 'steps': 32, 'york': 32, 'announcement':
31, 'available': 31, 'brian': 31, 'caused': 31, 'control': 31,
'estimates': 31, 'experience': 31, 'long': 31, 'penalties': 31,
'people': 31, 'trust': 31, 'vendor': 31, 'vice': 31, 'al.': 30,
'comprehensive': 30, 'conduct': 30, 'cong': 30, 'court': 30,
'encryption': 30, 'events': 30, 'full': 30, 'individuals': 30,
'knowledge': 30, 'last': 30, 'later': 30, 'magnetic': 30, 'making':
30, 'mobile': 30, 'multiple': 30, 'notice': 30, 'others': 30,
'strong': 30, 'verizon': 30
```

Output 6.2.3.5 presents a list of words, and the occurrence frequency count up to a count of 30 occurrences, in the Target corpus after the final training and conditioning process.

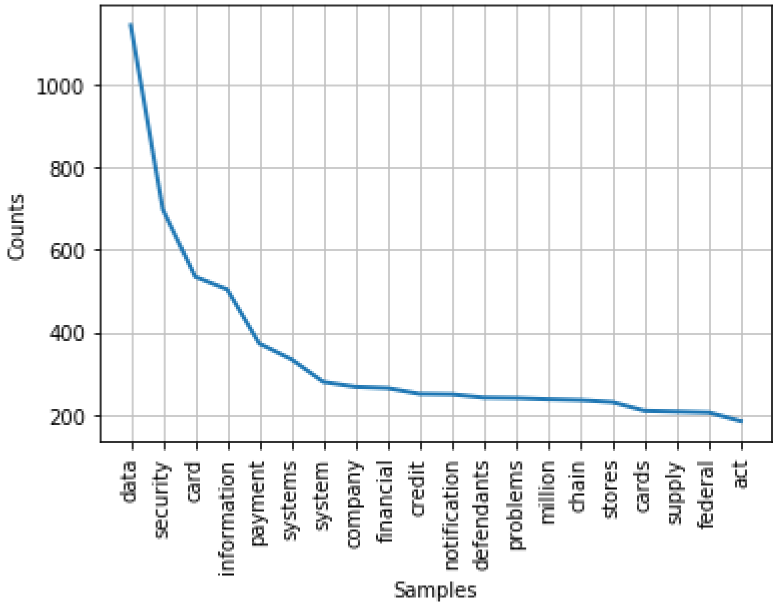
Input 6.2.3.7



```
freq_t_train = FreqDist(clean1)
freq_t_train.plot(20)
```

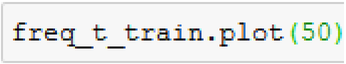
Input 6.2.3.6 calls for a frequency distribution of words contained within the clean1 tokenized file resulting from the training conditioning sequence. The call **FreqDist** provides the frequency distribution of the tokenized words output from Input 6.2.3.1, and saves the results as ‘freq_t_train’ The call **freq_t_train.plot(20)** presents a graph of the frequency count of the first 20 words contained within the conditioned file clean1.

Output 6.2.3.7



Output 6.2.3.7 shows a frequency count plot of the first 20 words, when categorized by frequency, contained within the Target corpora file, after training and labelled clean1. The plot output indicates that the word token ‘vehicle’ and ‘vehicles’ have been successfully removed.

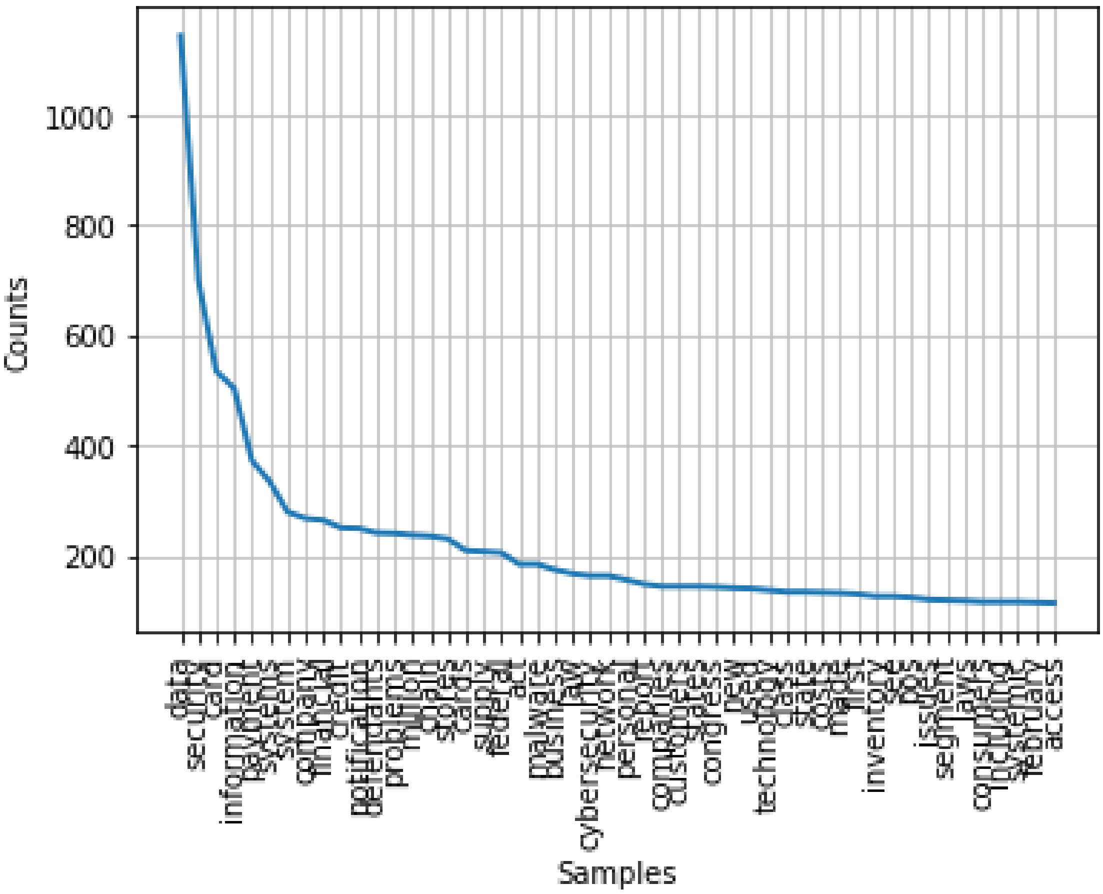
Input 6.2.3.8



```
freq_t_train.plot(50)
```

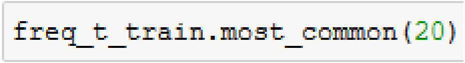
Input 6.2.3.8 uses **freq_t_train.plot(50)** to call for a graph of the frequency count of the first 50 words contained within the trained and conditioned file clean1.

Output 6.2.3.8



Output 6.2.3.8 shows a frequency count plot of the first 50 words, when categorized by frequency, contained within the Target corpora file ‘clean1’, after training and conditioning. The plot output indicates that the word token ‘target’ and ‘breach’ have been successfully removed.

Input 6.2.3.9



```
freq_t_train.most_common(20)
```

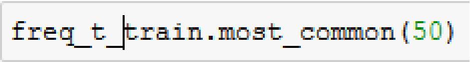
Input 6.2.3.9 calls for a list of the 20 most common words remaining within the Target corpus conditioned and trained file, identified by frequency, providing the associated word count for each tokenized word.

Output 6.2.3.9

```
('data', 1143),
('security', 698),
('card', 535),
('information', 504),
('payment', 373),
('systems', 335),
('system', 280),
('company', 268),
('financial', 265),
('credit', 251),
('notification', 250),
('defendants', 242),
('problems', 241),
('million', 238),
('chain', 236),
('stores', 231),
('cards', 210),
('supply', 208),
('federal', 206),
('act', 185)
```

Output 6.2.3.9 presents a word count of the 20 most common word occurrences contained within the Target conditioned and trained file, in list format.

Input 6.2.3.10



```
freq_t_train.most_common(50)
```

Input 6.2.3.10 calls for a list of the 50 most common words remaining within the Target corpus conditioned and trained file, identified by frequency, providing the associated word count for each tokenized word.

Output 6.2.3.10

```
('data', 1143),
('security', 698),
('card', 535),
('information', 504),
('payment', 373),
('systems', 335),
('system', 280),
('company', 268),
('financial', 265),
('credit', 251),
('notification', 250),
('defendants', 242),
('problems', 241),
('million', 238),
('chain', 236),
('stores', 231),
('cards', 210),
('supply', 208),
('federal', 206),
('act', 185),
('malware', 185),
('business', 175),
('law', 168),
('cybersecurity', 164),
('network', 164),
('personal', 157),
('report', 149),
('companies', 145),
('customers', 145),
('states', 145),
('congress', 144),
('new', 142),
('used', 141),
('technology', 138),
('class', 135),
('state', 135),
('costs', 134),
('made', 133),
('first', 131),
('inventory', 127),
('see', 127),
('pos', 125),
('issues', 121),
('segment', 120),
('laws', 119),
('consumers', 117),
('including', 117),
('systemic', 117),
('february', 116),
('access', 115)]
```

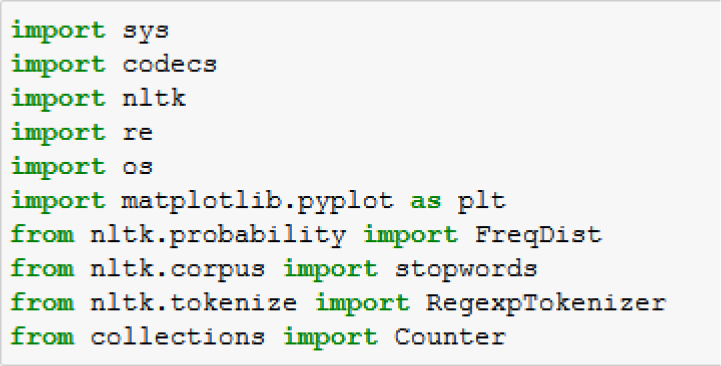
Output 6.2.3.10 shows the output of the 50 most common entries in the Target corpus after conditioning and training, along with the corresponding word-count value. The output represents the final SAE output to inform the following Maturity Analysis process.

## APPENDIX C.3 TESLA

The following sub-sections report the full data collection, the data processing, the visualization processes, and the algorithm training steps. These data methods are overviewed in Figure 1.1 and are consistent with automation of the expert system manually tested in the Pilot study. The computation of the maturity level values is then completed from these comprehensive outputs and reported in Chapter six.

**Process**

Input 6.3.1.1

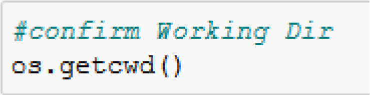


```
import sys
import codecs
import nltk
import re
import os
import matplotlib.pyplot as plt
from nltk.probability import FreqDist
from nltk.corpus import stopwords
from nltk.tokenize import RegexpTokenizer
from collections import Counter
```

Input 6.3.1.1 shows the packages and modules that will be required to perform the various functions within the semantic analysis engine. The **sys** module is imported to provide the Python interpreter information regarding constants, functions and methods, when using the ‘dir(system)’ call. The **codecs** module defines the base classes for standard Python encoders and decoders (codecs). The **codecs** module allow the Python interpreter to encode text to bytes and also encode text to text and bytes to bytes. The error handling lookup process is also managed by the **codecs** module. The **nltk** import command provides access to the Natural Language Tool Kit suite of text processing libraries. The libraries of nltk modules include the tokenization, word frequency and list of default stopwords. The **re** import command provides the Python interpreter regular expression matching operations. Both Unicode strings and 8-bit strings can be searched and matched. Import **os** allows the Python interpreter to interface with the Windows or Linux operating system, enabling cross platform functionality.

**matplotlib.pyplot** imports a command style function collection, providing a stateful plotting area in a graphing output, keeping track of current Figures and plotting area. The **matplotlib.pyplot** command has been imported as **plt** to enhance usability. **FreqDist** is imported as a function component of the **nltk.probability** module, allowing the Python interpreter to display the frequency distribution of each word within the selected corpora. Stopwords are imported from the **nltk.corpus** package of texts, providing a default list of **stopwords**, such as ‘and’, ‘if’, ‘then’ and ‘of’.

Input 6.3.1.2



```
#confirm Working Dir
os.getcwd()
```

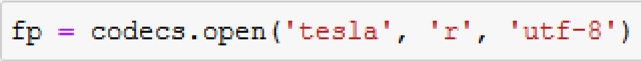
Input 6.3.1.2 Confirms the working directory. Each experiment is constrained to a separate and distinct directory. This assists with corpora delineation, allowing the selection of topic specific corpus.

Output 6.3.1.2

```
'C:\\Users\\User\\Desktop\\PhD\\CH6\\SAE\\Tesla'
```

Output 6.3.1.2 shows that the working directory is ‘tesla’ as part of the Semantic Analysis Engine folder.

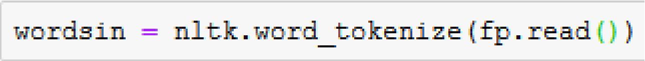
Input 6.3.1.3



```
fp = codecs.open('tesla', 'r', 'utf-8')
```

Input 6.3.1.3 queries the working director for the identified corpus, called’ tesla’, then open the corpus using the **codecs.open** call. The corpus will then open using the utf-8 formatting structure, and then label the output ‘fp’.

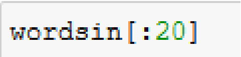
Input 6.3.1.4



```
wordsin = nltk.word_tokenize(fp.read())
```

Input 6.3.1.4 reads the corpus output from ‘Input 6.3.1.3’ labelled ‘fp’, and then tokenizes the results using the **‘nltk.word_tokenize’** call from the **nltk** module. Input 6.3.1.4 provides a tokenized output labelled ‘wordsin’, which word-breaks the entire corpus sentencing structure into single words.

Input 6.3.1.5



```
wordsin[:20]
```

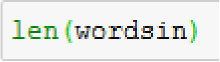
Input6.3.1.5 calls for the first 20 words from the tokenized stream labelled ‘wordsin’. The purpose of Input 6.3.1.5 is to confirm the word-breaking process has been implemented correctly, by retrieving the first 20 words of the Tesla corpus.

Output 6.3.1.5

```
'Assessment',
'and',
'standardization',
'of',
'autonomous',
'vehicles',
'Abstract',
'Autonomous',
'vehicle',
'technology',
'presents',
'a',
'huge',
'challenge',
'to',
'standardization',
'and',
'legal',
'bodies',
'.'
```

Output 6.3.1.5 shows the first 20 words of the Tesla corpus, correctly identifying each separate word, therefore showing that the word-breaking process has occurred.

Input 6.3.1.6



```
len(wordsin)
```

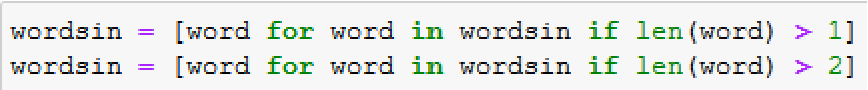
Input 6.3.1.6 calls for a count of the number of words contained within the word-broken output from Input 6.3.1.4 using **len.**

Output 6.3.1.6

```
67272
```

Output 6.3.1.6 provides a primary word count of the entire corpus, with no conditioning or training. The word count output allocates a baseline Figure that will be used to determine the effectiveness of each subsequent training of conditioning sequence.

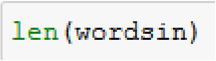
Input 6.3.1.7



```
wordsin = [word for word in wordsin if len(word) > 1]
wordsin = [word for word in wordsin if len(word) > 2]
```

Input 6.3.1.7 is the first of the conditioning sequences. Input 6.3.1.7 parses the ‘wordsin’ file and removes any word token that is less than 2 characters in length, providing an output that overwrites the ‘wordsin’ file. Input 6.3.1.7 also removes any punctuation that has been tokenized.

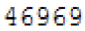
Input 6.3.1.8



```
len(wordsin)
```

Input 6.3.1.8 calls for a count of the number of words contained within the filtered output from the Input 6.3.1.7 process using **len.**

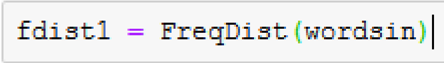
Output 6.3.1.8



```
46969
```

Output 6.3.1.8 shows that 20,303 tokenized words have been removed, with 46,969 tokenized words remaining, when compared to the output 6.3.1.6.

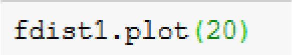
Input 6.3.1.9



```
fdist1 = FreqDist(wordsin)
```

Input 6.3.1.9 calls for an initial frequency distribution of words contained within the Tesla corpus. The call **FreqDist** provides the frequency distribution of the tokenized words output from 6.3.1.7, and saves the results as ‘fdist1’

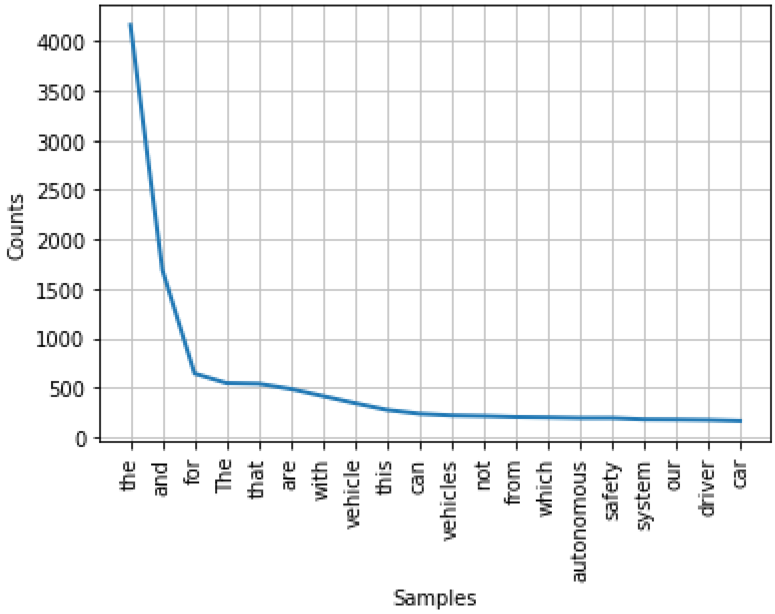
Input 6.3.1.10



```
fdist1.plot(20)
```

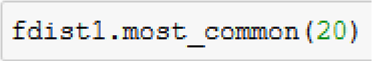
Input 6.3.1.10 calls for a plot graph of the first 20 words within the fdist1 file using **fdist1.plot**

Output 6.3.1.0



Output 6.3.1.10shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Tesla corpora. The initial plot indicates that there are word tokens that do not add any meaningful input.

Input 6.3.1.11



```
fdist1.most_common(20)
```

Input 6.3.1.11 calls for a list of the 20 most common words remaining within the Tesla corpus, identified by frequency, giving the associated wordcount for each tokenized word.

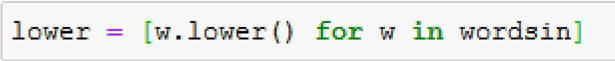
Output 6.3.1.11

```
('the', 4163),    ***
('and', 1687),    ***
('for', 641),     ***
('The', 544),     ***
('that', 538),    ***
('are', 485),     ***
```

```
('with', 413),    ***
('vehicle', 341),
('this', 273),    ***
('can', 234),     ***
('vehicles', 217),
('not', 211),     ***
('from', 201),    ***
('which', 197),   ***
('autonomous', 191),
('safety', 191),
('system', 177),
('our', 175),     ***
('driver', 171),
('car', 162)
```

Output 6.3.1.11 provides a word count of the most common word occurrences in list format. The output permits identification of words that do not add value to the experiment process. Output 6.3.1.11 indicates that there are 13 words that can be removed from the top 20 occurring words, without removing additional 'meaning' The identified word tokens are indicated with '***'. The out 6.3.1.11 list also shows that words beginning with a capital letter are distinct for words without a capital letter, as seen with the word 'The' and 'the' in the out 6.3.1.11 list.

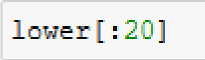
Input 6.3.1.12



```
lower = [w.lower() for w in wordsin]
```

Input 6.3.1.12 parses the tokenized wordlist using the call **lower()** to convert all the word tokens to lowercase. The output is saved as 'lower'

Input 6.3.1.13



```
lower[:20]
```

Input 6.3.1.13 calls for a list of the first 20 words in the 'lower' tokenized word stream, to validate the lowercase conversion process of in 6.3.1.12.

Output 6.3.1.13

```
'assessment',
'and',
'standardization',
'autonomous',
'vehicles',
'abstract',
'autonomous',
'vehicle',
'technology',
'presents',
'huge',
'challenge',
'standardization',
'and',
'legal',
'bodies',
'this',
'new',
'rapidly',
'emerging'
```

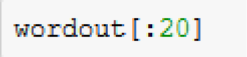
Output 6.3.1.13 demonstrates that the lower-case conversion process has been effective. The word ‘assessment’ is now lower-case, when compared to the output 6.3.1.5.

Input

```
wordout = sorted(lower)
```

Input 6.3.1.14 sorts the output shown in out 6.3.1.13 into alpha-numeric order, using the call **sorted**, and saves the output as ‘wordout’

Input 6.3.1.15



```
wordout[:20]
```

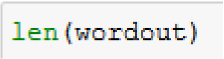
Input 6.3.1.15 calls for a list output of the first 20 entries in the alpha-numerically sorted file labelled wordout.

Output 6.3.1.15

```
'-nal',
'...',
'.al',
'//ieeexplore.ieee.org',
'//www.ieee.org/publications_standards/publications/rights/index
.html'
'//www.iso.org/iso/catalogue_detail',
'//www.rand.org/blog/articles/2017/11/',
'//zalazone.hu/en',
'/width',
'0-1',
'0.0005',
'0.007',
'0.01',
'0.01',
'0.01',
'0.01',
'0.01415',
'0.01991',
'0.03552',
'0.09751'
```

Output 6.3.1.15 shows that the first 20 entries all little value to the experiment. The output provides a list of conditioning vectors that will be addressed in each of the following steps.

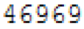
Input 6.3.1.16



```
len(wordout)
```

Input 6.3.1.16 calls for the number of tokenized words remaining in the Tesla corpus. As no words have been removed, the output is compared to out 6.3.1.8 to validate the corpus.

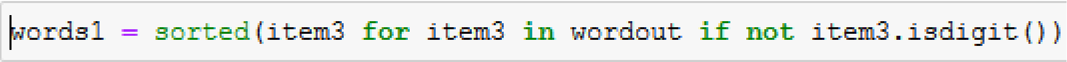
Output 6.3.1.16



```
46969
```

Output 6.3.1.16 is the same as out 6.3.1.8, showing that the conversion to lower-case has not removed any tokenized words from the Tesla corpus.

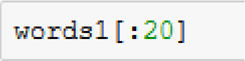
Input 6.3.1.17



```
words1 = sorted(item3 for item3 in wordout if not item3.isdigit())
```

Input 6.3.1.17 parses the 'wordout' file produced from 6.3.1.14, identifies and then removes any tokenized words that contain digits using the **isdigit** call. The output is then sorted into alpha-numeric format and then saved as 'words1'.

Input 6.3.1.18



```
words1[:20]
```

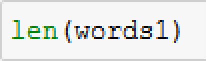
Input 6.3.1.18 calls for a list of the first 20 entries within the 'words1' tokenized and sorted

Output 6.3.1.18

```
'-nal',
'...',
'.al',
'//ieeexplore.ieee.org',
'//www.ieee.org/publications_standards/publications/rights/index.html'
'//www.iso.org/iso/catalogue_detail',
'//www.rand.org/blog/articles/2017/11/',
'//zalazone.hu/en',
'/width',
'0-1',
'0.0005',
'0.007',
'0.01',
'0.01',
'0.01',
'0.01',
'0.01415',
'0.01991',
'0.03552',
'0.09751'
```

Output 6.3.1.18 shows that entries that no digits have been removed, because any digit string that contains an entry that is not a digit, remains.

Input 6.3.1.19



```
len(words1)
```

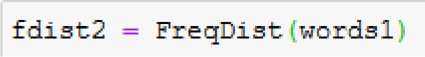
Input 6.3.1.19 calls for a word count of the number of tokenized entire remaining in the Tesla corpus.

Output 6.3.1.19

```
46678
```

Output 6.3.1.19 shows that 291 tokenized word entries have been removed from the Tesla corpus when compared to the output 6.3.1.16

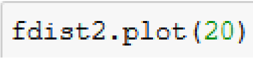
Input 6.3.1.20



```
fdist2 = FreqDist(words1)
```

Input 6.3.1.20 calls for a frequency distribution of words contained within the Tesla corpus after the first phase of the comprehensive conditioning sequence. The call **FreqDist** provides the frequency distribution of the tokenized words output from 6.3.1.17, and saves the results as 'fdist2'

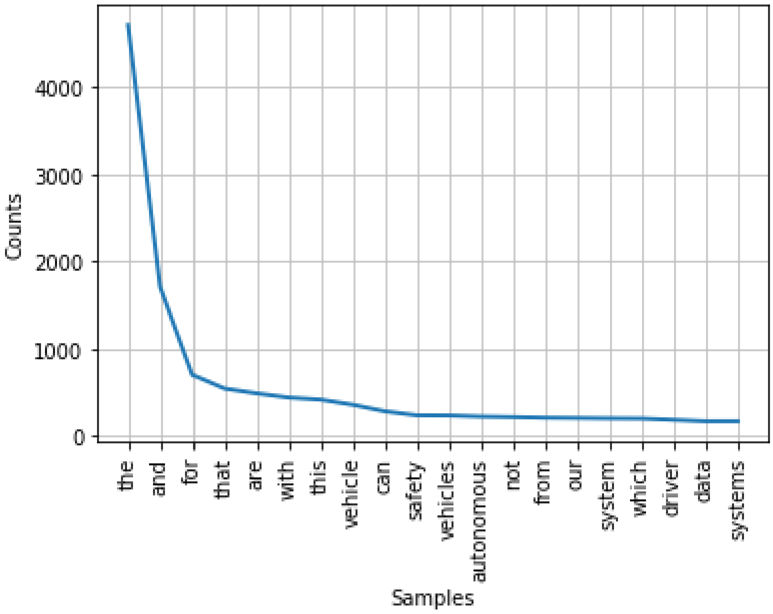
Input 6.3.1.21



```
fdist2.plot(20)
```

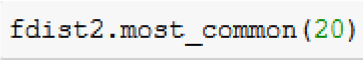
Input 6.3.1.21 calls for a plot graph of the first 20 words within the fdist2 file using **fdist2.plot**

Output6.3.1.21



Output 6.3.1.21 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Tesla corpus file labelled words1. The initial plot indicates that there are still word tokens remaining that do not add any meaningful input.

Input 6.3.1.22



```
fdist2.most_common(20)
```

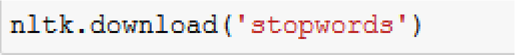
Input 6.3.1.22 calls for a list of the 20 most common words remaining within the Tesla corpus file labelled words1, identified by frequency, providing the associated word count for each tokenized word.

Output 6.3.1.22

```
('the', 4715), ***
('and', 1707), ***
('for', 700),  ***
('that', 543), ***
('are', 488),  ***
('with', 440), ***
('this', 416), ***
('vehicle', 357),
('can', 282),  ***
('safety', 236),
('vehicles', 234),
('autonomous', 222),
('not', 216),  ***
('from', 208), ***
('our', 205),  ***
('system', 201),
('which', 199),***
('driver', 184),
('data', 168),
('systems', 168)
```

Output 6.3.1.22 provides a word count of the most common word occurrences in list format. The output permits identification of words that remain after the first stage of conditioning that do not add value to the experiment process. Output 6.3.1.22 indicates that there are 12 words remaining that have not been removed from the top 20 occurring words. These word tokens are indicate with '***'. The out 6.3.1.22 list also shows that words beginning with a capital letter and the words without a capital letter, have been combined, as seen with the word 'the' in the out 6.3.1.22 list

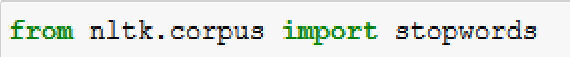
Input 6.3.1.23



```
nltk.download('stopwords')
```

6.3.1.23 downloads the package 'stopwords' from the nltk database. 6.3.1.23 downloads a corpora of stopwords in 16 different languages.

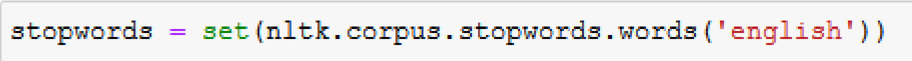
Input 6.3.1.24



```
from nltk.corpus import stopwords
```

Input 6.3.1.24 imports the stopword corpus, ready to be called by the following input 6.3.1.25

Input 6.3.1.25



```
stopwords = set(nltk.corpus.stopwords.words('english'))
```

Input 6.3.1.25 calls for the default stopword corpus containing English stopwords, and sets the output to a file labelled 'stopwords'

Input 6.3.1.26

```
print(stopwords)
```

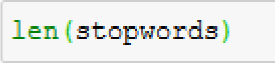
Input 6.3.1.26 calls for a list of the default stopwords contained within stopword file.

Output 6.3.1.26

```
'any', "you're", 'more', 'you', 'had', 'd', 'our', 'hasn',
'myself', "you'd", 'ma', 'haven', 'most', 'has', 'very',
'there', 'mustn', 'is', 'they', 'now', 'to', 'its', 'the',
'off', 'hadn', 'we', 'were', 'again', 'against', 'over', 'didn',
"she's", "isn't", 'not', 'by', 'what', 'so', 'hers', 'needn',
'it', "haven't", 'at', 'as', "mustn't", 'does', 've', 't',
"shan't", 'o', 'too', 'few', "couldn't", 'herself', 'same',
'in', "won't", 'are', 'will', 'shan', 'until', 'my', 's',
'wasn', 'because', 'own', 'she', 'after', 'isn', 'i', 'why',
'down', 'can', 'this', 'wouldn', 'on', 'only', "you'll", 'both',
"wouldn't", 'than', 'a', "weren't", 'who', 'being', 'which',
'such', 'below', "should've", 'weren', "needn't", 'out', 'for',
"doesn't", 'all', 'themselves', 'aren', 'himself', "it's",
'into', 'under', 'ourselves', 'have', "mightn't", 'those',
'itself', 'no', 'whom', 'up', 'll', "hadn't", 'how', "didn't",
'mightn', 'y', 'am', 'do', 'and', 'each', 'here', 'between',
'other', 'm', 'doesn', "you've", 'your', 'ain', 'just',
"that'll", 'doing', 'did', 'these', 'while', 'shouldn', 'me',
'through', 'above', 'their', "wasn't", 'him', 'that',
"shouldn't", "don't", 'of', 'he', 'further', 'if', 'been', 're',
'during', 'where', 'his', 'ours', 'be', 'having', 'her', 'once',
'them', 'then', 'nor', 'yourselves', 'some', 'yourself', 'but',
'with', 'yours', "hasn't", 'couldn', "aren't", 'should', 'was',
'theirs', 'or', 'before', 'from', 'won', 'an', 'don', 'when',
'about'
```

Output 6.3.1.26 displays the list of default English stopwords downloaded from the nltk database.

Input 6.3.1.27



```
len(stopwords)
```

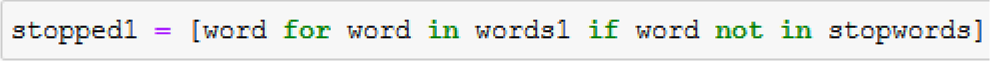
Input 6.3.1.27 calls for a count of the number of stopwords contained within the English stopword corpus.

Output 6.3.1.27

```
179
```

Output 6.3.1.27 shows that there are 179 words contained within the English stopword corpus.

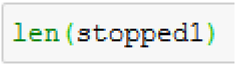
Input 6.3.1.28



```
stopped1 = [word for word in words1 if word not in stopwords]
```

Input 6.3.1.28 parses the words1 tokenized word file, resulting from 6.3.1.17, removing any token contained within the stopword file. The output tokenized file is saved as ‘stopped1’

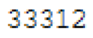
Input 6.3.1.29



```
len(stopped1)
```

Input 6.3.1.29 calls for a count of the entries contained within the ‘stopped1’ tokenized file, to validate that entries have been removed

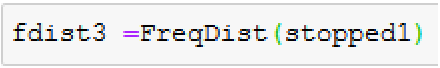
Output 6.3.1.29



```
33312
```

Output 6.3.1.29 shows that 13,366 entries have been removed after parsing for any English stopword contained within the default stopword file, when compared to the output 6.3.1.19.

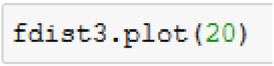
Input 6.3.1.30



```
fdist3 =FreqDist(stopped1)
```

Input 6.3.1.30 calls for a frequency distribution of words contained within the stopped1 tokenized file resulting from the first default stopword conditioning sequence. The call **FreqDist** provides the frequency distribution of the tokenized words output from Input 6.3.1.28, and saves the results as ‘fdist3’

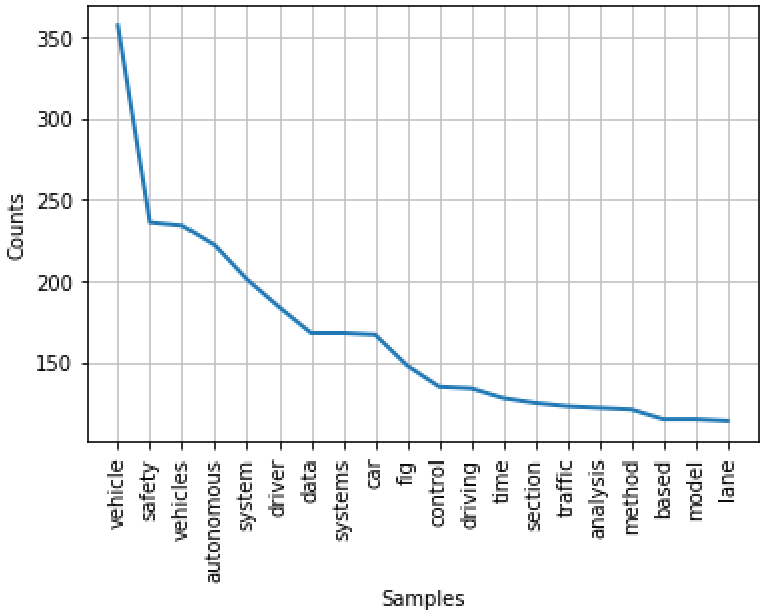
Input 6.3.1.31



```
fdist3.plot(20)
```

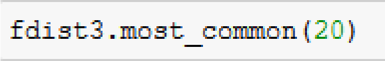
Input 6.3.1.31 calls for a plot graph of the first 20 words within the fdist3 file using **fdist3.plot**

Output 6.3.1.31



Output 6.3.1.31 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Tesla corpora file, after stopwords have been removed, labelled stopped1. The initial plot indicates that the stop word tokens have been successfully removed.

Input 6.3.1.32



```
fdist3.most_common(20)
```

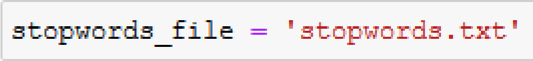
Input 6.3.1.32 calls for a list of the 20 most common words remaining within the Tesla corpus file labelled stopped1, identified by frequency, providing the associated word count for each tokenized word.

Output 6.3.1.32

```
('vehicle', 357),
('safety', 236),
('vehicles', 234),
('autonomous', 222),
('system', 201),
('driver', 184),
('data', 168),
('systems', 168),
('car', 167),
('fig', 148),
('control', 135),
('driving', 134),
('time', 128),
('Section', 125),
('traffic', 123),
('analysis', 122),
('method', 121),
('based', 115),
('model', 115),
('lane', 114)
```

Output 6.3.1.32 provides a word count of the most common word occurrences contained within the stopped1 file, in list format. The output permits identification of words that remain after the first default stoppword removal conditioning, that do not add value to the experiment process. Output 6.3.1.32 indicates that the 11 words that were identified as providing no value, as shown from out 6.3.1.22, are removed from the top 20 occurring words.

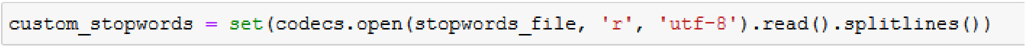
Input 6.3.1.33



```
stopwords_file = 'stopwords.txt'
```

Input 6.3.1.33 links an external text file called ‘stopwords.txt’ that has been created to provide a custom list of domain specific stopwords. The custom stopwords are stored in a text file with a blank first line and each word on a separate line thereafter.

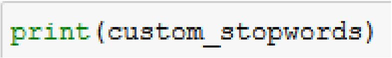
Input 6.3.1.34



```
custom_stopwords = set(codecs.open(stopwords_file, 'r', 'utf-8').read().splitlines())
```

Input 6.3.1.34 parses the text file using the **codecs.open** call, using a **splitlines** parameter that identifies each custom stopword on each new line. The parsed output is then saved as a file labelled custom_stopwords.

Input 6.3.1.35



```
print(custom_stopwords)
```

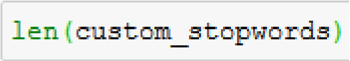
Input 6.3.1.35 calls for a list of the custom stopwords contained within the file 'custom_stopwords'

Output 6.3.1.35

```
'Table', '7', 'Figure', 'however', 'may', 'nine', 'six', '4',
'eight', 'one', '0', '1', 'fig', 'four', '\ufeff', '9',
'moreover', 'ten', '.', '5', 'would', 'two', '2', '6', 'also',
'five', 'finally', 'Section', '3', '8', 'therefore', 'seven',
'could', 'although', 'three'
```

Output 6.135 provides the list of custom stopwords.

Input 6.3.1.36



```
len(custom_stopwords)
```

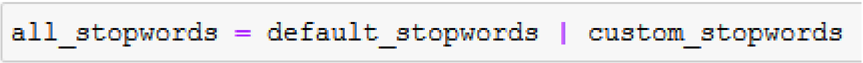
Input 6.3.1.36 calls for a count of the number of entries contained within the 'custom_stopwords' file

Output 6.3.1.36

```
35
```

Output 6.3.1.36 shows that there are 35 entries contained within the custom_stopwords file

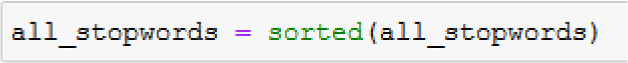
Input 6.3.1.37



```
all_stopwords = default_stopwords | custom_stopwords
```

Input 6.3.1.37 merges the default_stopwords file with the custom_stopwords into a new file labelled all_stopwords

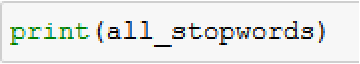
Input 6.3.1.38



```
all_stopwords = sorted(all_stopwords)
```

Input 6.3.1.38 sorts all the entries in the all_stopwords file into alpha-numeric order.

Input 6.3.1.39



```
print(all_stopwords)
```

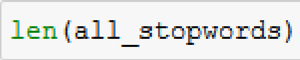
Input 6.3.1.39 calls for a list of all the entries contained within the merged all_stopwords file.

Output 6.3.1.39

```
'.', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a',
'about', 'above', 'after', 'again', 'against', 'ain', 'all',
'also', 'although', 'am', 'an', 'and', 'any', 'are', 'aren',
"aren't", 'as', 'at', 'be', 'because', 'been', 'before',
'being', 'below', 'between', 'both', 'but', 'by', 'can',
'could', 'couldn', "couldn't", 'd', 'did', 'didn', "didn't",
'do', 'does', 'doesn', "doesn't", 'doing', 'don', "don't",
'down', 'during', 'each', 'eight', 'few', 'fig', 'Figure',
'finally', 'five', 'for', 'four', 'from', 'further', 'had',
'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven',
"haven't", 'having', 'he', 'her', 'here', 'hers', 'herself',
'him', 'himself', 'his', 'how', 'however', 'i', 'if', 'in',
'into', 'is', 'isn', "isn't", 'it', "it's", 'its', 'itself',
'just', 'll', 'm', 'ma', 'may', 'me', 'mightn', "mightn't",
'more', 'moreover', 'most', 'mustn', "mustn't", 'my', 'myself',
'needn', "needn't", 'nine', 'no', 'nor', 'not', 'now', 'o',
'of', 'off', 'on', 'once', 'one', 'only', 'or', 'other', 'our',
'ours', 'ourselves', 'out', 'over', 'own', 're', 's', 'same',
'Section', 'seven', 'shan', "shan't", 'she', "she's", 'should',
"should've", 'shouldn', "shouldn't", 'six', 'so', 'some',
'such', 't', 'Table', 'ten', 'than', 'that', "that'll", 'the',
'their', 'theirs', 'them', 'themselves', 'then', 'there',
'therefore', 'these', 'they', 'this', 'those', 'three',
'through', 'to', 'too', 'two', 'under', 'until', 'up', 've',
'very', 'was', 'wasn', "wasn't", 'we', 'were', 'weren',
"weren't", 'what', 'when', 'where', 'which', 'while', 'who',
'whom', 'why', 'will', 'with', 'won', "won't", 'would',
'wouldn', "wouldn't", 'y', 'you', "you'd", "you'll", "you're",
"you've", 'your', 'yours', 'yourself', 'yourselves', '\ufeff'
```

Output 6.3.1.39 provides an alpha-numeric list of the merged all_stopwords file

Input 6.3.1.40



```
len(all_stopwords)
```

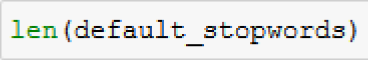
Input 6.3.1.40 calls for a count of all the entries contained within the merged all_stopwords file

Output 6.3.1.40

```
214
```

Output 6.3.1.40 provides a count of 214 entries in the merged all_stopwords file.

Input 6.3.1.41



```
len(default_stopwords)
```

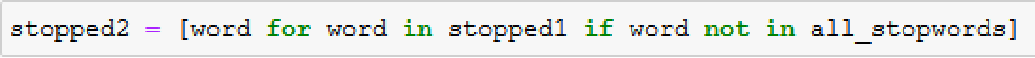
Input 6.3.1.41 calls for a count of all the entries contained within the default_stopwords file

Output 6.3.1.41

```
179
```

Output 6.3.1.41 provides a count of 179 entries contained within the default_stopwords file, indicating that there have been 35 words added, and providing validation confirmation that the custom stop words have been added as required.

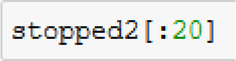
Input 6.3.1.42



```
stopped2 = [word for word in stopped1 if word not in all_stopwords]
```

Input 6.3.1.42 parses the stopped1 tokenized word file, resulting from 6.3.1.28, removing any token contained within the custom all_stopwords file. The output tokenized file is saved as 'stopped2

Input 6.3.1.45



```
stopped2[:20]
```

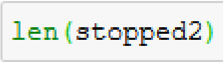
Input 6.3.1.45 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file 'stopped2'

Output 6.3.1.45

```
'-nal',
'...',
'.al',
'//ieeexplore.ieee.org',
'//www.ieee.org/publications_standards/publications/rights/index
.html'
'//www.iso.org/iso/catalogue_detail',
'//www.rand.org/blog/articles/2017/11/',
'//zalazone.hu/en',
'/width',
'0-1',
'0.0005',
'0.007',
'0.01',
'0.01',
'0.01',
'0.01',
'0.01415',
'0.01991',
'0.03552',
'0.09751'
```

Output 6.3.1.45 provides a list of the first 20 alpha-numeric tokenized entries contained within the file 'stopped2'. The list output indicates that there all the first 20 entries identified do not add value to the data, and can therefore be removed from the file.

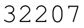
Input 6.3.1.46



```
len(stopped2)
```

Input 6.3.1.46 calls for a count of the number of entries contained within the 'stopped2' file

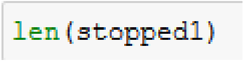
Output 6.3.1.46



```
32207
```

Output 6.3.1.46 shows that the stopped2 file contains 17,507 tokenized entries

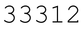
Input 6.3.1.47



```
len(stopped1)
```

Input 6.3.1.47 calls for a count of the number of entries contained within the ‘stopped1’ file

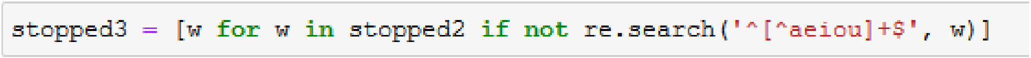
Output 6.3.1.47



```
33312
```

Output 6.3.1.47 shows that the stopped1 file contains 33,312 tokenized entries, indicating that 1,105 tokenized entries have been removed when applying the custom_stopwords contained within the all_stopwords file.

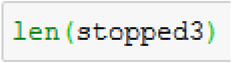
Input 6.3.1.48



```
stopped3 = [w for w in stopped2 if not re.search('^[^aeiou]+$', w)]
```

Input 6.3.1.48 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped2’ file that does not contain a vowel. The output is saved as a tokenized file labelled ‘stopped3’

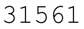
Input 6.3.1.49



```
len(stopped3)
```

Input 6.3.1.49 calls for a count of the number of entries contained within the ‘stopped3’ file

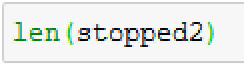
Output 6.3.1.49



```
31561
```

Output 6.3.1.49 shows that the stopped3 file contains 31,561 tokenized entries within the ‘stopped3’ file

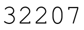
Input 6.3.1.50



```
len(stopped2)
```

Input 6.3.1.50 calls for a count of the number of entries contained within the ‘stopped2’ file

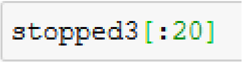
Output 6.3.1.50



```
32207
```

Output 6.3.1.50 shows that the ‘stopped2’ file contains 32,207 entries, indicating that 646 entries have been removed when applying the regular expression provided by 6.3.1.48 to remove any entry that does not contain a vowel.

Input 6.3.1.51



```
stopped3[:20]
```

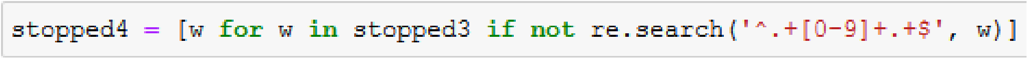
Input 6.3.1.51 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file ‘stopped3’

Output 6.3.1.51

```
'-nal',
'.al',
'//ieeexplore.ieee.org',
'//www.ieee.org/publications_standards/publications/rights/index
.html'
'//www.iso.org/iso/catalogue_detail',
'//www.rand.org/blog/articles/2017/11/',
'//zalazone.hu/en',
'/width',
'0tu',
'10.1109/mcom.2018.1700884',
'10fi4',
'10fi5',
'10fi56',
'10fi6',
'11-bit',
'114e282',
'11a',
'1609.2a-2017',
'1here',
'20-car'
```

Output 6.3.1.51 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped3’. The list output indicates that there are entries that do not add value to the data, and can therefore be removed from the file.

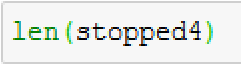
Input 6.3.1.52



```
stopped4 = [w for w in stopped3 if not re.search('^.+[0-9]+.+$', w)]
```

Input 6.3.1.52 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped3’ file that contains a number. The output is saved as a tokenized file labelled ‘stopped4’

Input 6.3.1.53



```
len(stopped4)
```

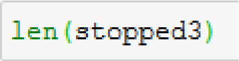
Input 6.3.1.53 calls for a count of the number of entries contained within the ‘stopped4’ file

Output 6.3.1.53

```
31498
```

Output 6.3.1.53 shows that the stopped4 file contains 31,498 tokenized entries

Input 6.3.1.54



```
len(stopped3)
```

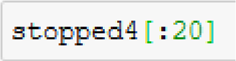
Input 6.3.1.54 calls for a count of the number of entries contained within the 'stopped3' file

Output 6.3.1.54

```
31561
```

Output 6.3.1.54 shows that the' stopped3' file contains 31,561 tokenized entries, indicating that 63 entries have been removed when applying the regular expression provided by 6.3.1.52 to remove any entry that contains a number.

Input 6.3.1.55



```
stopped4[:20]
```

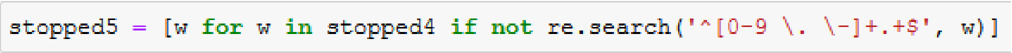
Input 6.3.1.55 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file 'stopped4'

Output 6.3.1.55

```
'-nal',           ***
'.al',            ***
'//ieeexplore.ieee.org', ***
'//www.ieee.org/publications_standards/publications/rights/index
.html'            ***
'//www.iso.org/iso/catalogue_detail',   ***
'//zalazone.hu/en',      ***
'/width',         ***
'0tu',            ***
'1here',          ***
'2the',           ***
'3-lane',         ***
'3two',           ***
'4measured',      ***
'5we',            ***
'6we',            ***
'7note',          ***
'8the',           ***
'abbreviated',
'abbreviations',
'abilities'
```

Output 6.3.1.55 provides a list of the first 20 alpha-numeric tokenized entries contained within the file 'stopped4'. The list output indicates that there is at least one entry that does not add value to the data, and can therefore be removed from the file. The entries identified are highlighted.

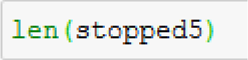
Input 6.3.1.56



```
stopped5 = [w for w in stopped4 if not re.search('^[0-9 \. \-]+.+$', w)]
```

Input 6.3.1.56 utilizes a regular expression to identify and remove any tokenized entry in the 'stopped4' file that begins with a number. The output is saved as a tokenized file labelled 'stopped5'

Input 6.3.1.57



```
len(stopped5)
```

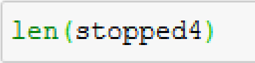
Input 6.3.1.57 calls for a count of the number of entries contained within the 'stopped5' file

Output 6.3.1.57

```
31486
```

Output 6.3.1.57 shows that the stopped5 file contains 31,486 tokenized entries

Input 6.3.1.58



```
len(stopped4)
```

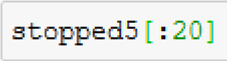
Input 6.3.1.58 calls for a count of the number of entries contained within the 'stopped4' file

Output 6.3.1.58

```
31498
```

Output 6.3.1.58 shows that the 'stopped4' file contains 31,498 entries, indicating that 12 entries have been removed when applying the regular expression provided by 6.3.1.56 to remove any entry that begins with a number.

Input 6.3.1.59



```
stopped5[:20]
```

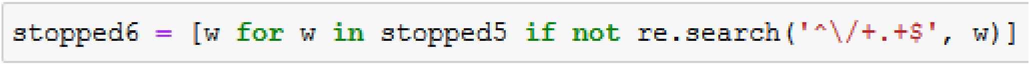
Input 6.3.1.59 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file 'stopped5'

Output 6.3.1.59

```
'//ieeexplore.ieee.org', ***
'//www.ieee.org/publications_standards/publications/rights/index
.html'           ***
'//www.iso.org/iso/catalogue_detail',   ***
'//zalazone.hu/en',      ***
'/width',        ***
'abbreviated',
'abbreviations',
'abilities',
'abilities',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability'
```

Output 6.3.1.59 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped5’. The list output indicates that there are entries that do not add value to the data, and can therefore be removed from the file. The entries identified are identified by ‘***’.

Input 6.3.1.60



```
stopped6 = [w for w in stopped5 if not re.search('^\/+.+$', w)]
```

Input 6.3.1.60 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped5’ file that begins with a ‘forward slash’ as determined by a hyperlink to a URL. The output is saved as a tokenized file labelled ‘stopped6’

Input 6.3.1.61

```
len(stopped6)
```

Input 6.3.1.61 calls for a count of the number of entries contained within the ‘stopped6’ file

Output 6.3.1.61

```
31481
```

Output 6.3.1.61 shows that the ‘stopped6’ file contains 31,481 tokenized entries

Input 6.3.1.62

```
len(stopped5)
```

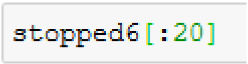
Input 6.3.1.62 calls for a count of the number of entries contained within the ‘stopped5’ file

Output 6.3.1.62

```
31486
```

Output 6.3.1.62 shows that the ‘stopped5’ file contains 31,484 entries, indicating that 5 entries have been removed when applying the regular expression provided by 6.3.1.60 to remove any entry that indicates a URL.

Input 6.3.1.63



```
stopped6[:20]
```

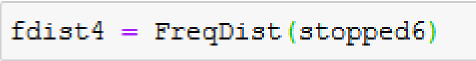
Input 6.3.1.63 calls for a list of the first 20 entries contained within the alpha-numerically sorted, tokenized word file ‘stopped6’

Output 6.3.1.63

```
'abbreviated',
'abbreviations',
'abilities',
'abilities',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'ability',
'able',
'able',
'able'
```

Output 6.3.1.63 provides a list of the first 20 alpha-numeric tokenized entries contained within the file ‘stopped6’. The list output indicates that all entries, within the first 20 identified, add value to the data.

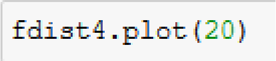
Input 6.3.1.64



```
fdist4 = FreqDist(stopped6)
```

Input 6.3.1.64 calls for a frequency distribution of words contained within the stopped6 tokenized file resulting from the conditioning sequences. The call **FreqDist** provides the frequency distribution of the tokenized words contained within stopped6, and saves the results as ‘fdist4’

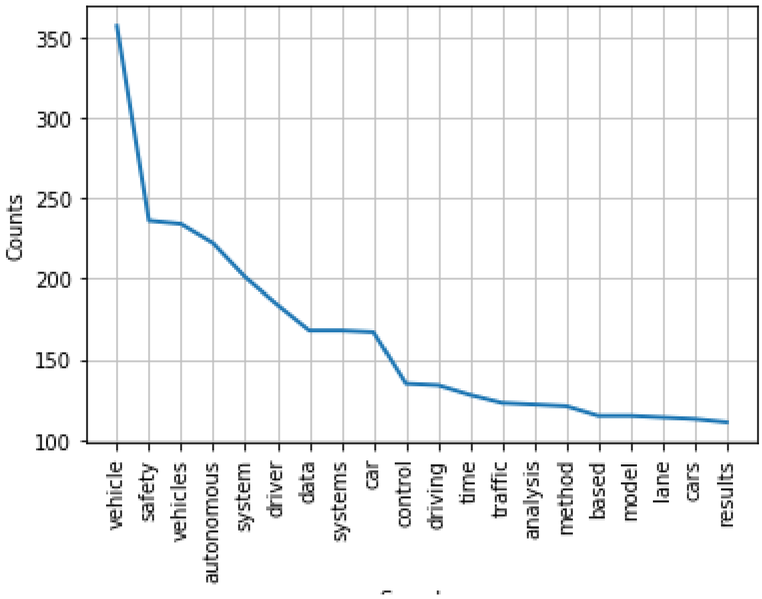
Input 6.3.1.65



```
fdist4.plot(20)
```

Input 6.3.1.31 calls for a plot graph of the first 20 words within the fdist4 file using **fdist4.plot**

Output 6.3.1.65



Output 6.3.1.65 shows a frequency plot of the first 20 words, when categorized by frequency, contained within the Tesla corpora file, after the conditioning process, labelled fdist4.

Input 6.3.1.66

```
freq4.most_common(50)
```

Input 6.3.1.66 calls for the top 50 entries contained within the ‘stopped6’ tokenized file.

Output 6.3.1.66

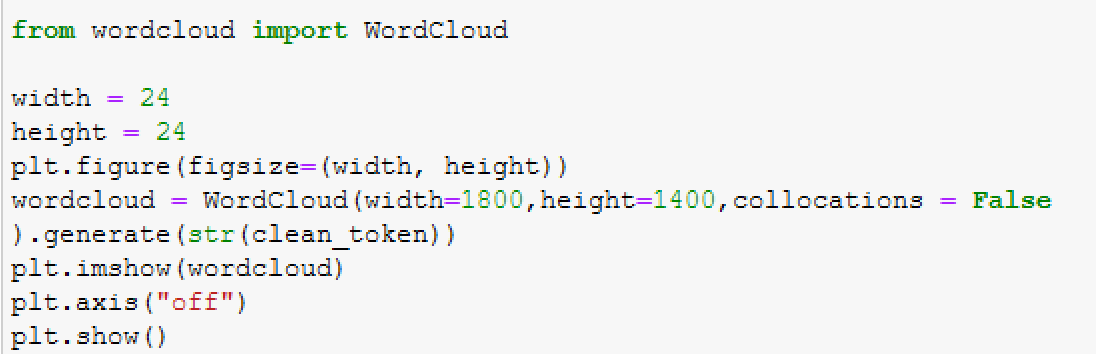
```
('vehicle', 357),
('safety', 236),
('vehicles', 234),
('autonomous', 222),
('system', 201),
('driver', 184),
('data', 168),
('systems', 168),
('car', 167),
('control', 135),
('driving', 134),
('time', 128),
('traffic', 123),
('analysis', 122),
('method', 121),
('based', 115),
('model', 115),
('lane', 114),
('cars', 113),
('results', 111),
('information', 108),
('used', 106),
('detection', 103),
('study', 103),
('automotive', 102),
('test', 102),
('human', 99),
('using', 97),
('number', 96),
('level', 95),
('testing', 93),
('case', 91),
('participants', 87),
('see', 85),
('standard', 82),
('use', 82),
('paper', 81),
('manufacturers', 79),
('drivers', 78),
('iso', 76),
('risk', 76),
('software', 76),
('environment', 74),
('hazard', 74),
('proposed', 71),
('step', 69),
('process', 68),
('distance', 67),
('road', 67),
('different', 66)
```

Output 6.3.1.66 shows the output of the 50 most common entries, along with the corresponding word-count value of ‘stopped6’. The file ‘stopped 6’ is the final tokenized output of the stopword conditioning sequences.

### WordCloud Visualisation

The next step is to implement further training and conditioning processing. The output of the training process presents conditioned data, to inform the final risk maturity enumeration. The output from Section 6.3.1 is exhibited in the form of a WordCloud visualization. Using the Word Cloud output as a visualisation tool, Output 6.3.2.1 shows that the word ‘vehicle’ has been identified as the word that occurs most frequently. As the word ‘vehicle does not add benefit to the risk identification process, the word is a good example to demonstrate the training abilities of the prototype. The training presents a conditioned output, that allows further reduction of ‘noise’ words, as identified in the Section 6.3.1 process.

Input 6.3.2.1

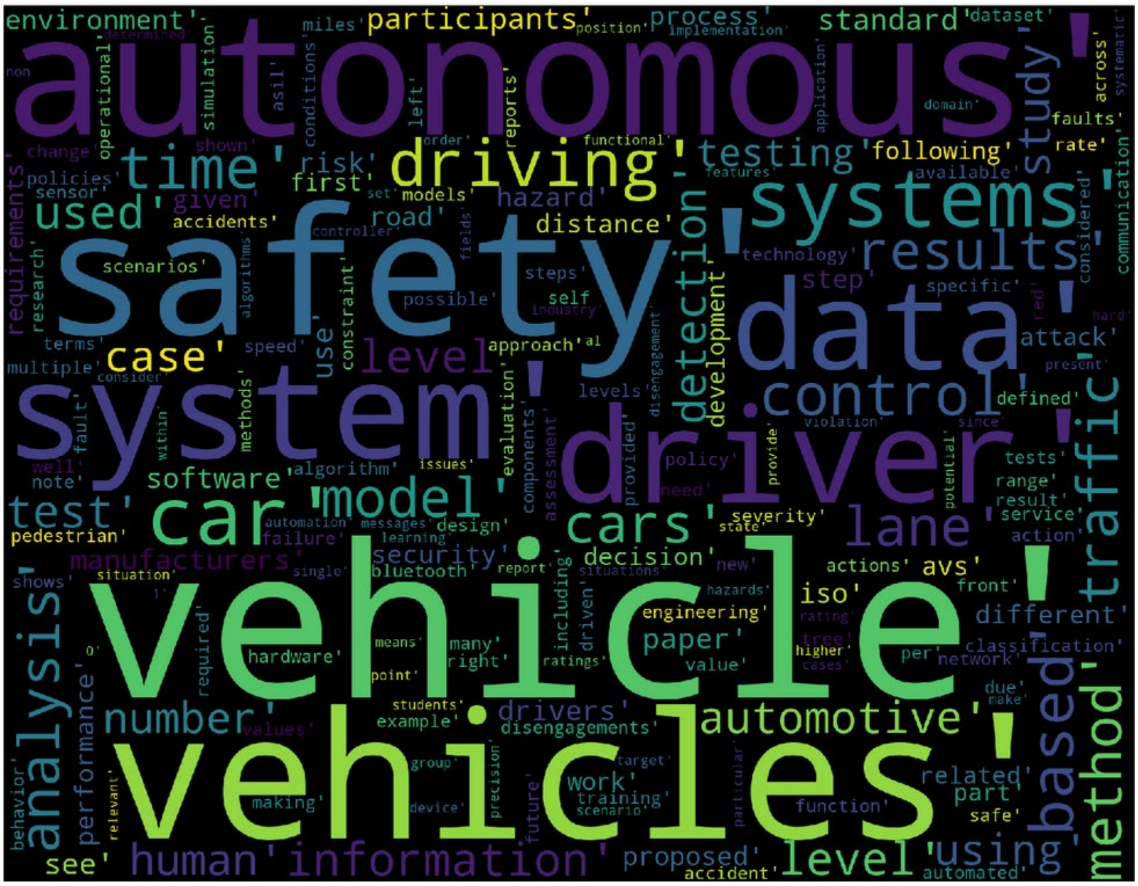


```
from wordcloud import WordCloud

width = 24
height = 24
plt.figure(figsize=(width, height))
wordcloud = WordCloud(width=1800,height=1400,collocations = False
).generate(str(clean_token))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Input 6.3.2.1 calls for the WordCloud module and defines the arguments to be implemented. The arguments define parameters of size, position and figuresize.

Output 6.3.2.1

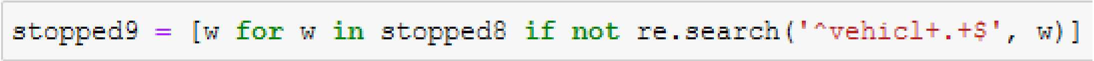


The WordCloud visualization presented as Output 6.3.2.1 shows that the words ‘vehicle’ is identified as the word that occurs most frequently. The visualization also shows that the word ‘vehicles’ also occurs frequently

**Training**

The output from the WordCloud visualization (Output 6.3.2.1) shows that the words ‘vehicle’ and ‘vehicles’ are identified as ‘noise’ and can be removed from the word frequency output. The final output presents a trained word frequency list that is conditioned to better inform the final step of maturity modelling.

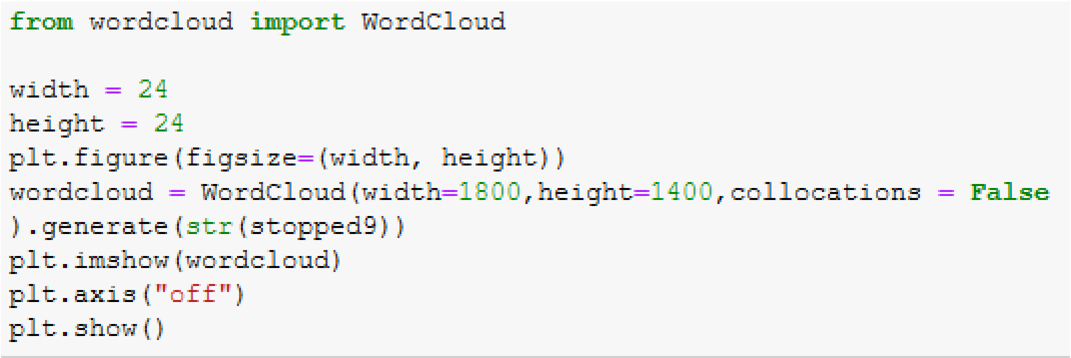
Input 6.3.3.1



```
stopped9 = [w for w in stopped8 if not re.search('^vehicl+.+$', w)]
```

Input 6.3.3.1 utilizes a regular expression to identify and remove any tokenized entry in the ‘stopped8’ file that contains the word vehicle, which includes the words ‘vehicle’ and ‘vehicles’. The output is saved as a tokenized file labelled ‘stopped9’

Input 6.3.3.2

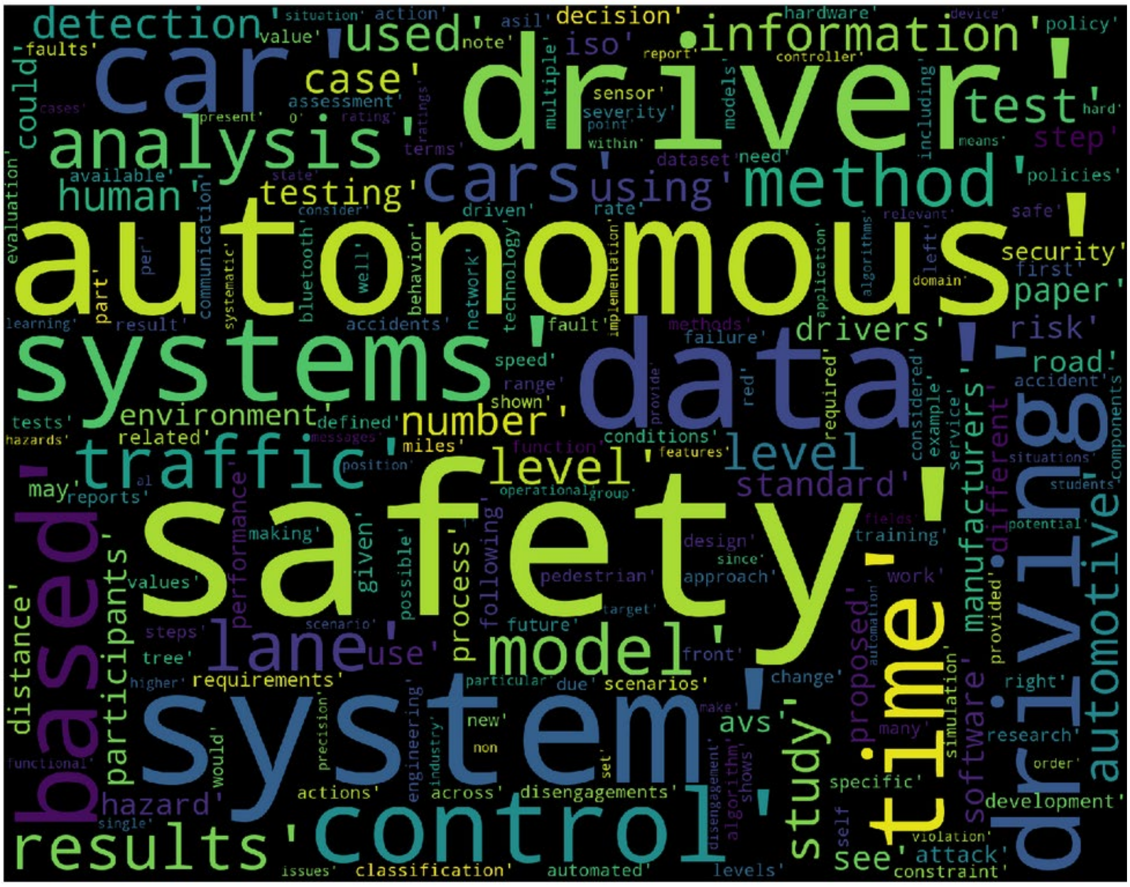


```
from wordcloud import WordCloud

width = 24
height = 24
plt.figure(figsize=(width, height))
wordcloud = WordCloud(width=1800,height=1400,collocations = False
).generate(str(stopped9))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

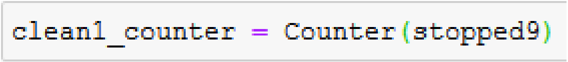
Input 6.3.3.2 resets the WordCloud module and calls for a new visualization using the ‘stopped9’ tokenized file. The output is utilized to determine whether the training process has been effective, and presents a conditioned output.

Output 6.3.3.2



The WordCloud visualization presented as Output 6.3.3.2 shows that the word ‘vehicle’ is no longer present. The visualization demonstrates the tokenized output is conditioned and provides an effective input for the final maturity analysis process.

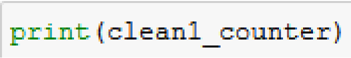
Input 6.3.3.3



```
clean1_counter = Counter(stopped9)
```

The file ‘stopped9’ is processed with a counter algorithm using Input 6.3.3.3, that counts the frequency occurrence of each word within the trained and conditioned file, and the output is labelled clean1_counter.

Input 6.3.3.4



```
print(clean1_counter)
```

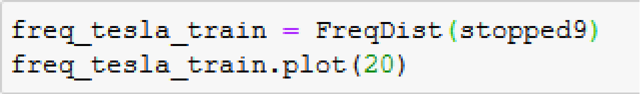
Input 6.3.3.4 calls for a list of the words contained within the conditioned file ‘clean1_counter’ with the corresponding occurrence frequency count.

Output 6.3.3.4

```
'safety': 236, 'autonomous': 222, 'system': 201, 'driver': 184,
'data': 168, 'systems': 168, 'car': 167, 'control': 135, 'driving':
134, 'time': 128, 'traffic': 123, 'analysis': 122, 'method': 121,
'based': 115, 'model': 115, 'lane': 114, 'cars': 113, 'results': 111,
'information': 108, 'used': 106, 'detection': 103, 'study': 103,
'automotive': 102, 'test': 102, 'human': 99, 'using': 97, 'number':
96, 'level': 95, 'testing': 93, 'case': 91, 'participants': 87, 'see':
85, 'standard': 82, 'use': 82, 'paper': 81, 'manufacturers': 79,
'drivers': 78, 'iso': 76, 'risk': 76, 'software': 76, 'environment':
74, 'hazard': 74, 'proposed': 71, 'step': 69, 'process': 68,
'distance': 67, 'road': 67, 'different': 66, 'security': 66,
'performance': 65, 'decision': 63, 'given': 63, 'attack': 61,
'requirements': 60, 'avs': 59, 'following': 59, 'work': 59,
'development': 57, 'first': 57, 'part': 57, 'disengagements': 56,
'example': 55, 'methods': 55, 'policies': 55, 'asil': 54, 'actions':
52, 'severity': 52, 'accidents': 51, 'pedestrian': 51, 'due': 50,
'function': 50, 'across': 49, 'assessment': 49, 'policy': 49, 'well':
49, 'change': 48, 'classification': 47, 'fault': 47, 'future': 47,
'miles': 47, 'faults': 46, 'levels': 46, 'training': 46, 'design': 45,
'rate': 45, 'algorithm': 44, 'engineering': 44, 'new': 44, 'safe': 44,
'approach': 43, 'per': 43, 'range': 43, 'technology': 43, 'dataset':
42, 'multiple': 42, 'specific': 42, 'failure': 41, 'provided': 41,
'related': 41, 'scenarios': 41, 'tree': 41, 'value': 41, 'available':
40, 'communication': 40, 'constraint': 40, 'driven': 40, 'left': 40,
'result': 40, 'shown': 40, 'shows': 40, 'tests': 40, 'action': 39,
'hardware': 39, 'many': 39, 'models': 39, 'network': 39, 'speed': 39,
'terms': 39, 'values': 39, 'accident': 38, 'conditions': 38,
'including': 38, 'need': 38, 'possible': 38, 'red': 38, 'steps': 38,
'sensor': 37, 'service': 37, 'automated': 36, 'note': 36, 'bluetooth':
35, 'front': 35, 'reports': 35, 'required': 35, 'research': 35,
'right': 35, 'simulation': 35, 'behavior': 34, 'considered': 34,
'defined': 34, 'evaluation': 34, 'implementation': 34, 'present': 34,
'students': 34, 'within': 34, 'domain': 33, 'features': 33, 'issues':
33, 'make': 33, 'messages': 33, 'operational': 33, 'particular': 33,
'point': 33, 'relevant': 33, 'since': 33, 'situations': 33, 'state':
33, 'violation': 33, 'al.': 32, 'components': 32, 'position': 32,
'precision': 32, 'rating': 32, 'report': 32, 'situation': 32,
'systematic': 32, 'consider': 31, 'device': 31, 'fields': 31,
'functional': 31, 'hard': 31, 'higher': 31, 'industry': 31, 'level-0':
31, 'level-1': 31, 'order': 31, 'potential': 31, 'scenario': 31,
'set': 31, 'algorithms': 30, 'provide': 30, 'self-driving': 30
```

Output 6.3.3.4 presents a list of the first 50 words, along with the associated word frequency count, up to a count of 30 occurrences, in the Tesla corpus after the final training and conditioning process.

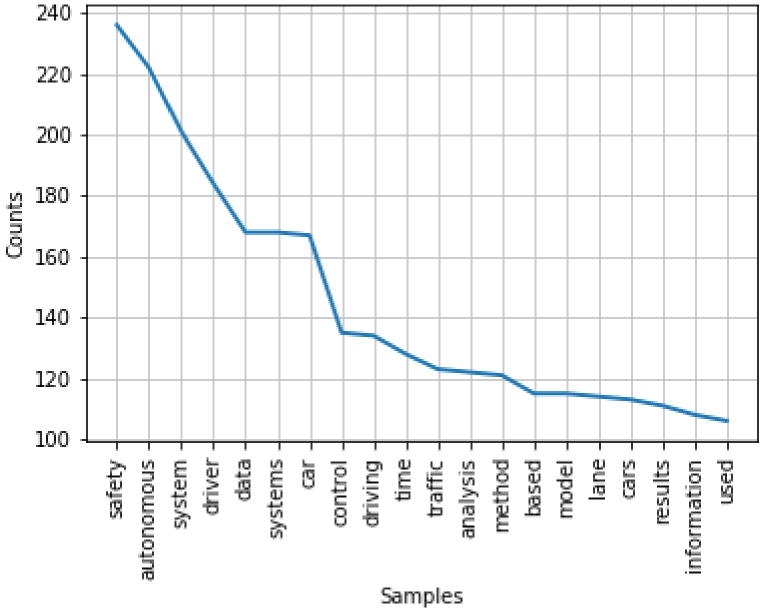
Input 6.3.3.5



```
freq_tesla_train = FreqDist(stopped9)
freq_tesla_train.plot(20)
```

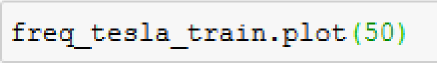
Input 6.3.3.5 calls for a frequency distribution of words contained within the stopped9 tokenized file resulting from the training conditioning sequence. The call **FreqDist** provides the frequency distribution of the tokenized words output from Input 6.3.3.1, and saves the results as ‘freq_tesla_train’ The call **freq_tesla_train.plot(20)** presents a graph of the frequency count of the first 20 words contained within the conditioned file stopped9.

Output 6.3.3.5



Output 6.3.3.5 shows a frequency count plot of the first 20 words, when categorized by frequency, contained within the Tesla corpora file, after training and, labelled stopped9. The plot output indicates that the word token ‘vehicle’ and ‘vehicles’ have been successfully removed.

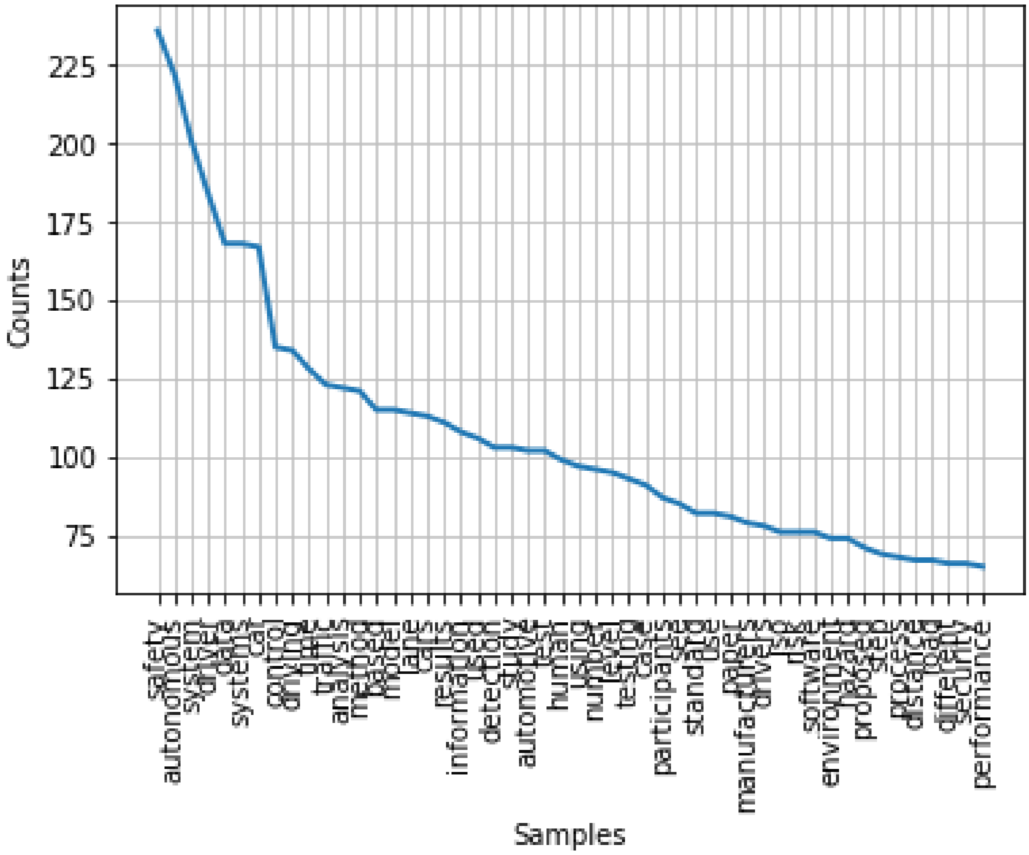
Input 6.3.3.6



```
freq_tesla_train.plot(50)
```

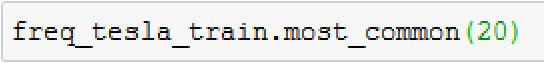
Input 6.3.3.6 uses **freq_tesla_train.plot(50)** to call for a graph of the frequency count of the first 50 words contained within the conditioned file stopped9.

Output 6.3.3.6



Output 6.3.3.6 shows a frequency count plot of the first 50 words, when categorized by frequency, contained within the Tesla corpora file, after training and labelled stopped9. The plot output indicates that the word token ‘vehicle’ and ‘vehicles’ have been successfully removed.

Input 6.3.3.7



```
freq_tesla_train.most_common(20)
```

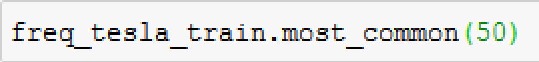
Input 6.3.3.7 calls for a list of the 20 most common words remaining within the Tesla corpus conditioned and trained file, identified by frequency, providing the associated word count for each tokenized word.

Output 6.3.3.7

```
('safety', 236),
('autonomous', 222),
('system', 201),
('driver', 184),
('data', 168),
('systems', 168),
('car', 167),
('control', 135),
('driving', 134),
('time', 128),
('traffic', 123),
('analysis', 122),
('method', 121),
('based', 115),
('model', 115),
('lane', 114),
('cars', 113),
('results', 111),
('information', 108),
('used', 106)]
```

Output 6.3.3.7 presents a word count of the most common word occurrences contained within the Tesla conditioned and trained file, in list format.

Input 6.3.3.8



```
freq_tesla_train.most_common(50)
```

Input 6.3.3.8 calls for a list of the 50 most common words remaining within the Tesla corpus conditioned and trained file, identified by frequency, providing the associated word count for each tokenized word.

Output 6.3.3.8

```
('safety', 236),
('autonomous', 222),
('system', 201),
('driver', 184),
('data', 168),
('systems', 168),
('car', 167),
('control', 135),
('driving', 134),
('time', 128),
('traffic', 123),
('analysis', 122),
('method', 121),
('based', 115),
('model', 115),
('lane', 114),
('cars', 113),
('results', 111),
('information', 108),
('used', 106),
('detection', 103),
('study', 103),
('automotive', 102),
('test', 102),
('human', 99),
('using', 97),
('number', 96),
('level', 95),
('testing', 93),
('case', 91),
('participants', 87),
('see', 85),
('standard', 82),
('use', 82),
('paper', 81),
('manufacturers', 79),
('drivers', 78),
('iso', 76),
('risk', 76),
('software', 76),
('environment', 74),
('hazard', 74),
('proposed', 71),
('step', 69),
('process', 68),
('distance', 67),
('road', 67),
('different', 66),
('security', 66),
('performance', 65
```

Output 6.3.3.8 presents

Output 6.3.3.8 shows the output of the 50 most common entries in the Tesla corpus after conditioning and training, along with the corresponding word-count value. The output represents the final SAE output to inform the following Maturity Analysis process.