

An Ontology-Driven AI Framework for Context-Aware Threat Detection in Mobile Health Applications

RAIYAN FERDOUS

A thesis submitted to the graduate faculty of Design and Creative Technologies

Auckland University of Technology
in partial fulfilment of the
requirements for the degree of
Master of Cyber Security and Digital Forensics

School of Engineering, Computer and Mathematical Sciences

Auckland, New Zealand
2026

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.

.....
RAIYAN FERDOUS

Acknowledgements

This research was completed as part of the Master of Cyber Security and Digital Forensics at the School of Computing and Mathematical Sciences in the Faculty of Design and Creative Technologies at AUT University in New Zealand. I would like to thank my parents for their incredible love and support throughout my entire academic life.

I am deeply grateful to my thesis supervisor, Dr Krassie Petrova, who has guided this work over the past year and a half. She has been everything one could hope for in a supervisor, and this thesis would not have been possible without her patience, insight, and encouragement.

I would also like to thank Dr Alistair Nisbet for his help and support throughout the entire master's program. My thanks to all the teachers, supervisors, and administrative staff at the School of Computing and Mathematical Sciences, who have been extraordinarily helpful in bringing this work to completion.

Finally, I would like to thank my peers in the Master of Cyber Security and Digital Forensics program for their support and friendship over the past 1.5 years.

Abstract

Mobile health (mHealth) applications have substantially improved the accessibility of healthcare, enabling remote monitoring, chronic-disease management, and personalized interventions. However, because these applications collect and transmit sensitive personal health information on resource-constrained mobile devices, they introduce significant security and privacy risks, including weak authentication, unsafe data sharing, and a lack of awareness of user context (Luxton et al., 2012; Aljedaani & Babar, 2021). Traditional, signature-based security methods are largely static and frequently fail to adapt to the changing situations in which mobile users operate, leaving context-dependent threats poorly addressed.

This thesis proposes a context-aware mHealth security architecture and develops an OWL 2 DL contextual ontology in Protégé. It separately evaluates a candidate machine-learning (ML) detection component on a network-intrusion benchmark. The architectural integration is a design proposition; its effect on detection performance has not been tested.

The empirical work consists of a Python/scikit-learn classification pipeline applied to NSL-KDD network-connection records, following the publicly available Kaggle notebook by josiagiven (n.d.). The ontology is a design artefact. The Android context-collection layer, ontology-to-ML transformation, mitigation module, and on-device application were designed but not implemented or empirically evaluated. To preserve participant privacy and satisfy ethical requirements, no real patient data were used.

The appendix records point estimates from a validation split, including Decision Tree accuracy of 0.983, KNN accuracy of 0.995, and Linear SVC accuracy of 0.981; the displayed Random Forest confusion matrix contains no errors and reports AUC = 1.00. These unusually high scores require verification for leakage and proxy features before being interpreted as reliable out-of-sample performance. The different ranges in Table 4.4 are retained as originally reported but cannot be identified as estimates from that same run without the fold-level records.

Latency below 150 ms and CPU utilization below 12% were proposed design targets. No Android deployment or device resource measurement was carried out.

The contribution is a specified ontology and mHealth security architecture together with an offline benchmark evaluation of a candidate ML detection layer. Whether semantic enrichment improves detection or an on-device implementation is feasible remains open to empirical testing.

Table of Contents

Contents

Abstract	4
Attestation of Authorship	7
1.1 Motivation for Research	8
1.2 Background	9
1.3 Problem Statement.....	11
1.4 Structure of Thesis.....	14
Chapter 2	17
Introduction to the Literature Review	17
2.1 Mobile Health Applications and Security Risks	20
2.2 Context-Aware Technologies	22
2.3 Ontology-Based Modeling.....	24
2.4 Integration of Ontology and Machine Learning.....	26
2.5 Current Challenges	27
2.6 Current Challenges and Research Gaps	28
Chapter 3: Research Methodology	31
3.0 Introduction	31
3.1 Research Design	32
3.2 Data Sources and Preparation.....	33
3.3 Prototype Architecture.....	34
3.4 Ontology Development Method	37
3.5 Machine Learning Methods.....	38
3.6 Experimental Procedure	39
3.7 Ethical Considerations	40
3.8 Limitations of the Method	41
3.9 Conclusion.....	42
4.0 Introduction	43
4.1 Prototype Development	44
4.1.3 Prototype components and implementation details	46
4.2 Experimental Setup	48
4.3 Data Generation and Collection.....	51

4.4 Evidence Examination and Analysis	53
4.5 Presentation of the Findings	56
4.6 Comparative Analysis and Evaluation.....	58
4.7 Conclusion.....	59
5.0 Introduction	60
Appendix 1: Raw Modelling and Simulation Output.....	60
Appendix 2.....	64
Artificial intelligence Declaration.....	64
5.1 Research Questions	67
5.2 Discussion of Findings	69
5.3 Comparison with the Literature	70
5.4 Limitations	71
5.5 Recommendations	72
5.6 Conclusion.....	72
Chapter 6.....	74
Conclusions, Limitations, and Future Research.....	74
6.0 Introduction	74
6.1 Summary of Research.....	74
6.1.1 Objectives and Hypotheses Revisited.....	74
6.2 Contributions of the Research	75
6.3 Limitations	76
6.4 Recommendations for Future Research	77
6.5 Final Remarks	77
References	78

Attestation of Authorship

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 (except where explicitly defined in the acknowledgements), nor used artificial intelligence tools or generative artificial intelligence tools (unless it is clearly stated, and referenced, along with the purpose of use), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Raiyan Ferdous

Chapter 1

Introduction

1.1 Motivation for Research

The motivation for this research stems from the pressing need to fortify mobile health (mHealth) applications against escalating cybersecurity threats in an era of pervasive digital health integration. As mHealth ecosystems expand—projected to reach USD 367.92 billion globally by 2033 with a compound annual growth rate (CAGR) of 17.4%—they have become indispensable for remote patient monitoring, chronic disease management, and teleconsultations, particularly amid rising chronic conditions like diabetes and cardiovascular diseases (Saxon, 2016). However, this growth amplifies vulnerabilities: mHealth devices, often resource-constrained wearables and smartphones, process sensitive personal health information (PHI) such as biometric data, geolocation, and behavioral patterns over unsecured networks, mirroring the openness and extensiveness of Internet of Things (IoT) architectures (Zaman et al., 2021). Traditional security measures, including firewalls and static encryption, falter against dynamic, context-dependent attacks like location spoofing or data exfiltration, leading to breaches that erode patient trust and incur substantial economic losses—estimated at over USD 6 trillion annually in global cybercrime by 2021 (Das & Sandhane, 2021).

A core driver is the inadequacy of conventional intrusion detection systems (IDS) in handling mHealth's fluid environments. In analogous domains like vehicle-to-grid (V2G) networks, where electric vehicles exchange energy data via Dedicated Short Range Communications (DSRC), attackers exploit mobility and limited ranges to launch denial-of-service (DoS) or man-in-the-middle (MitM) assaults, compromising confidential details such as charging times and locations (Saxena et al., 2016). Similarly, mHealth apps face DoS disruptions during critical monitoring sessions, with false positives from rule-based IDS causing alert fatigue and false negatives missing subtle anomalies (Al-Saleh, 2023). The 2023 Data Breach Statistics report over 14 billion records compromised since 2013, underscoring how zero-day attacks—evolving in sophistication—overwhelm signature-based defenses (Azam et al., 2023). This research is motivated by the imperative to transcend these limitations through ontology-driven artificial intelligence (AI), enabling semantic context modeling (e.g., inferring "high-risk public Wi-Fi access" from GPS and network data) fused with machine learning (ML) for adaptive anomaly detection (Das & Sandhane, 2021).

Furthermore, the integration of AI in cybersecurity addresses the heterogeneity and scalability challenges in mHealth, akin to IoT's distributed nodes. IoT devices, constrained by computing power, are prime targets for malware and eavesdropping, necessitating lightweight, intelligent countermeasures (Zaman et al., 2021). ML techniques like support vector machines (SVM) and decision trees (DT) have shown promise in IDS, achieving up to 98.5% accuracy in detecting network intrusions by refining feature selection via Gini impurity measures (Al-Saleh, 2023). Yet, standalone ML struggles with high-dimensional PHI data, prone to noise and evasion—evidenced in online social networks (OSNs) where instantaneous sharing of photos and videos invites cyberbullying or grooming, with over 70% of users reporting privacy breaches (Jain et al., 2021). Ontologies provide a structured knowledge representation to enrich ML inputs, reducing ambiguity and enhancing interpretability, as demonstrated in AI applications for network security where

hybrid models outperform traditional methods by 15–20% in recall (Perez Veiga, 2018). This thesis is compelled by the gap: While AI enhances IDS in general cybersecurity (Das & Sandhane, 2021), mHealth-specific fusions of ontologies with unsupervised ML (e.g., One-Class SVM) for on-device, privacy-preserving detection remain underexplored, despite regulatory mandates like the U.S. FDA's 2011 guidance emphasizing risk-benefit ratios for mobile medical apps (Saxon, 2016).

Regulatory and ethical requirements motivate the design of explainable, privacy-preserving mHealth security. This study responds by specifying an ontology-driven Android architecture and evaluating a separate candidate ML component. Detection thresholds and mobile resource limits are treated as future design and testing requirements rather than achieved outcomes.

Broader societal impacts drive the urgency. Cyberattacks in healthcare surged 45% post-2020, costing USD 20 billion in U.S. ransomware alone, disrupting services and endangering lives (Das & Sandhane, 2021). In IoT-integrated mHealth, such as wearable ECG monitors, undetected anomalies could delay interventions, echoing OSN's rapid misinformation spread (Jain et al., 2021). Reinforcement from analogous fields—AI in V2G for fine-grained access control—highlights transferable innovations: Blockchain-AI hybrids for key management in cyber-physical systems ensure tamper-proof session keys, adaptable to mHealth's edge computing (Saxena et al., 2016; Attkan & Ranga, 2022, as cited in prior works). Yet, challenges like data imbalance and adversarial evasion persist, motivating a design-science approach to validate hypotheses on ontology-ML synergy (Perez Veiga, 2018).

Economically, secure mHealth could save USD 300 billion annually in U.S. healthcare by reducing readmissions 38% via reliable monitoring (Saxon, 2016). This research is thus motivated by the dual promise of innovation and equity: Empowering low-resource settings with lightweight defenses, as in IoT's global deployments, while advancing AI ethics through interpretable models (Zaman et al., 2021). By addressing literature gaps—few studies integrate ontologies with ML for mobile-specific threats (Das & Sandhane, 2021; Azam et al., 2023)—it positions mHealth as a resilient ecosystem, fostering trust and adoption.

In summary, mHealth security motivates a contextual design, while the experiment in this thesis evaluates only the separate network-flow ML detection component.

1.2 Background

The rapid proliferation of mobile health (mHealth) applications has fundamentally transformed healthcare delivery in the 21st century. These applications, installed on smartphones and wearable devices, enable users to monitor vital signs, track chronic conditions, receive medication reminders, consult remotely with healthcare providers, and access personalized health information in real time. By leveraging ubiquitous mobile connectivity, sensors, and cloud integration, mHealth extends healthcare beyond traditional clinical settings, empowering patients with greater autonomy while supporting preventive care, remote monitoring, and timely interventions (Barton, 2012; Saxon, 2016). This shift aligns with broader societal demands for accessible, patient-centered care, particularly in an era marked by aging populations, rising chronic disease prevalence, and the lingering impacts of global health crises like the COVID-19 pandemic.

The global mHealth apps market has experienced explosive growth. As of 2025, the market was valued at approximately USD 40–43 billion, with projections indicating expansion to USD 113–134 billion by 2033–2034, reflecting compound annual growth rates (CAGRs) of 11.8–15.2% (Saxon, 2016). Health and fitness apps alone surpassed 3.6 billion downloads in recent years, while the broader health app sector generated billions in revenue, driven by rising smartphone penetration, aging populations, and the post-pandemic acceleration of digital health adoption. In the United States, the mHealth apps segment reached around USD 12–14 billion in 2024–2025, expected to exceed USD 32–37 billion by the early 2030s. This surge reflects not only technological maturity but also societal shifts toward patient-centered, accessible care models, particularly for managing chronic diseases such as diabetes, cardiovascular conditions, and mental health disorders (Barton, 2012). For instance, apps facilitating remote patient monitoring have demonstrated efficacy in reducing hospital readmissions by up to 38% for heart failure patients, underscoring mHealth's potential to alleviate overburdened healthcare systems (Saxon, 2016).

At its core, mHealth embodies the convergence of wireless technologies and clinical practice, defined as "the use of wireless communication devices to support public health and clinical practice" (Barton, 2012, p. 1). Mobile devices—encompassing smartphones, personal digital assistants, and wearable sensors—form the backbone of this ecosystem, enabling functionalities from appointment reminders and treatment adherence prompts to real-time biometric surveillance and epidemic outbreak tracking (Barton, 2012). The Royal Tropical Institute identifies eight key mHealth application areas: education and awareness systems for health promotion; point-of-care support and diagnostics for clinician decision-making; patient monitoring for adherence; disease surveillance; emergency response alerts; health information management; mobile learning (mLearning) for professionals; and financing tools like smart card payments (Barton, 2012). This diversity has democratized healthcare, bridging gaps in low- and middle-income countries where over 85% of the global population now has access to commercial wireless signals, supported by more than 5 billion mobile subscriptions (Barton, 2012). In resource-constrained settings, mHealth has advanced infrastructure development, outpacing traditional investments in roads and electricity by enabling remote diagnostics and community health worker coordination (Barton, 2012).

Yet, this transformative potential is tempered by profound security and privacy vulnerabilities. mHealth applications handle exquisitely sensitive data—personal health records, geolocation traces, biometric identifiers, and behavioral patterns—often transmitted over unsecured networks like public Wi-Fi. Traditional security paradigms, rooted in static rules and signature-based detection, falter in this dynamic milieu. As early as the mid-1990s, researchers highlighted the risks in mobile networking protocols, such as Mobile IP, where packet headers inadvertently expose user locations, enabling traffic analysis to construct invasive profiles of communication patterns, interests, and movements (Fasbender et al., 1996). This "ubiquitous computing" conundrum persists: mobile users demand seamless access to services "anywhere, anytime," yet systems often neglect encryption or contextual safeguards, as evidenced by lingering gaps in on-line banking security in regions like Germany (Fasbender et al., 1996). In mHealth, these lapses amplify risks, including unauthorized access, data exfiltration, identity theft, and inference attacks that correlate seemingly innocuous data (e.g., location and timestamp) to reveal health statuses or habits.

Contemporary threats are multifaceted and context-dependent. For example, location spoofing—altering GPS data to mimic legitimate access—can enable unauthorized data retrieval from public hotspots, while

behavioral anomalies, such as irregular medication adherence patterns, might signal device compromise or insider threats (Saxon, 2016). Regulatory bodies recognize these hazards: In July 2011, the U.S. Food and Drug Administration (FDA) issued draft guidance targeting mobile medical applications that function as accessories to medical devices or transform platforms into diagnostic tools, emphasizing patient safety amid innovation (Barton, 2012). Vos and Parker (as cited in Barton, 2012) argue that medical devices inherently pose hazards, necessitating regulations to mitigate risks through appropriate risk-benefit ratios. However, enforcement remains uneven; while low-risk apps (e.g., fitness trackers) evade scrutiny, high-stakes ones (e.g., cardiac monitors) demand rigorous validation, creating a fragmented landscape that stifles discovery (Barton, 2012). Moreover, evolving delivery models—exemplified by retail giants like CVS, Walgreens, and Target integrating onsite clinics with companion apps—exacerbate exposure, blending healthcare with consumer data ecosystems vulnerable to breaches (Saxon, 2016).

Privacy erosion extends beyond direct breaches. Traffic analysis in mobile IP protocols allows third parties to infer "who communicated with whom, at which place, and for how long," potentially compiling location profiles that undermine user autonomy (Fasbender et al., 1996, p. 1). In mHealth, this manifests as "context-insensitive" controls: static permissions fail to adapt to fluid scenarios, such as a user accessing sensitive records from a high-risk public network versus a secure home environment. Resultant issues include alert fatigue from excessive false positives, missed subtle threats (e.g., low-intensity data leaks), and eroded trust, deterring adoption—critical when 82% of consumers express interest in nontraditional services but cite privacy as a barrier (Saxon, 2016). Analogous challenges in adjacent domains, like vehicle-to-grid (V2G) networks, underscore the pattern: dynamic, data-rich systems demand fine-grained authentication, anonymity, and attestation to protect identities amid mobility and resource constraints (Saxena et al., 2016; Mohammadzadeh Mianji et al., 2025). In V2G, for instance, electric vehicles' bidirectional energy flows expose confidential details (e.g., charging times, locations) to eavesdroppers, necessitating privacy-preserving payments and access controls—mirroring mHealth's needs for secure, adaptive defenses (Saxena et al., 2016).

Conventional mitigations—coarse-grained permissions, rule-based firewalls, and basic encryption—prove inadequate against evolving attacks. Signature-dependent intrusion detection overlooks zero-day exploits, while cloud-centric models introduce latency and offload risks, contravening privacy-by-design principles (Barton, 2012; Fasbender et al., 1996). The literature reveals persistent gaps: ontologies for semantic context modeling exist in cybersecurity but rarely integrate with machine learning (ML) for mHealth-specific anomaly detection (Fasbender et al., 1996). ML techniques like One-Class Support Vector Machines (SVM) excel at unsupervised outlier identification, yet without semantic enrichment, they grapple with noisy, high-dimensional sensor data (Saxon, 2016). Reinforcement learning (RL) shows promise in adaptive trust management for vehicular networks, dynamically optimizing policies against threats, but its application to mHealth remains underexplored (Mohammadzadeh Mianji et al., 2025).

1.3 Problem Statement

The explosive growth of mobile health (mHealth) applications has revolutionized healthcare accessibility, yet it has simultaneously amplified critical security and privacy vulnerabilities that threaten patient trust, data integrity, and regulatory compliance. mHealth apps, which leverage smartphones, wearables, and wireless connectivity to deliver remote monitoring, chronic disease management, medication adherence

support, teleconsultations, and personalized health insights, process vast quantities of highly sensitive personal health information (PHI) (Barton, 2012; Saxon, 2016). This includes biometric data (e.g., heart rate, glucose levels), geolocation traces, behavioral patterns, and treatment histories—often transmitted over insecure networks and stored on resource-constrained devices. While these tools empower users and alleviate healthcare system burdens, the inherent dynamism of mobile environments—marked by variable locations, intermittent connectivity, and frequent context shifts—exposes them to sophisticated threats that static, traditional security mechanisms cannot adequately address (Fasbender et al., 1996; Saxena et al., 2016).

Market expansion underscores the urgency: The global mHealth market was valued at approximately USD 81.71–101.96 billion in 2025, with projections ranging from USD 268.46 billion by 2034 (CAGR 14.13%) to USD 367.92 billion by 2033 (CAGR 17.4%), driven by smartphone ubiquity, chronic disease prevalence, and post-pandemic telehealth acceleration (Saxon, 2016). In the United States, the segment alone is anticipated to surpass USD 100 billion by the mid-2030s. This proliferation democratizes care—particularly in low- and middle-income regions where wireless coverage exceeds 85% globally—but simultaneously expands the attack surface (Barton, 2012). Apps now facilitate everything from real-time epidemic surveillance to point-of-care diagnostics, yet they operate in heterogeneous ecosystems akin to the Internet of Things (IoT), where resource-constrained nodes (e.g., wearables) interconnect via protocols like Bluetooth Low Energy or Wi-Fi, inheriting IoT's openness and extensiveness (Wu et al., 2020; Zaman et al., 2021). The expansion of mHealth increases the number of devices, networks and interactions involving sensitive health information. This section therefore focuses on threat mechanisms and the need for contextual interpretation, rather than repeating market forecasts from the introduction.

These vulnerabilities are not generic IoT weaknesses; they attach to the specific data, sensors, and protocols that mHealth apps depend on, and they recur across the three architectural layers through which an app handles PHI: perception, network, and application. At the perception layer, the entry points are the wearable and smartphone sensors that feed remote-monitoring and medication-adherence apps — Bluetooth Low Energy (BLE) heart-rate straps, continuous glucose monitors, and accelerometers. Because BLE pairings are frequently established without authenticated encryption, an adversary within radio range can mount a man-in-the-middle attack to read or alter biometric data in transit, so that a glucose value or an arrhythmia flag arriving at the app is no longer trustworthy; a tampered glucose reading can in turn drive an erroneous insulin-dosing reminder (Karygiannis & Owens, 2002; Chan & Perrig, 2003). In the body area networks that wearable apps form, the same low-cost, low-power sensors that make deployment cheap also make physical node capture economically feasible, exposing on-device cryptographic keys to wider infiltration (Gao et al., 2018).

At the network layer the threat is concrete because the protocols are. mHealth apps move PHI over IEEE 802.11 Wi-Fi, Bluetooth, and cellular GPRS — all broadcast-oriented and often unencrypted above the link layer. On open Wi-Fi, passive packet capture with commodity tools such as Wireshark can expose not only credentials but adherence behaviour: *when* and *where* a patient opens a medication app lets a remote observer infer a chronic condition from the pattern of access rather than its content (Karygiannis & Owens, 2002). In the low-connectivity settings where mHealth is most valuable, GPRS interruptions — a median of 1.5 per person-month in rural HIV-adherence monitoring — leave PHI queued on the device or in the network, opening windows for denial-of-service and replay of that queued data, while the SMS+GPRS fallback that restores

connectivity itself admits SMS-spoofing of adherence and emergency alerts (Siedner et al., 2012). The same GPS stream that lets a health app route a user to the nearest pharmacy becomes, once intercepted, a means of inferring health status from location — proximity to an oncology clinic or a substance-treatment provider — without ever reading a clinical record (Saxon, 2016).

At the application layer the exposure is documented rather than hypothetical: more than 70% of health apps fail basic security audits, commonly lacking end-to-end encryption and embedding third-party advertising SDKs that harvest device identifiers alongside PHI (Boulos et al., 2014). Sideloaded or poorly vetted apps are a malware vector — the 2017 WannaCry campaign that encrypted patient records illustrates the ransomware end-state — and "anonymised" aggregate exports from fitness and wellness apps have repeatedly been re-identified by correlation with public records, breaching GDPR's data-minimisation principle (Boulos et al., 2014; Marin, 2005). Where apps integrate with online social networks, instantaneous PHI sharing widens the surface to social engineering and the grooming of vulnerable users, while wearables conscripted into botnets can mount distributed denial-of-service floods that disrupt teleconsultation (Jain et al., 2021).

What distinguishes these mHealth vulnerabilities from the generic catalogue is their *context-dependence*, and this is precisely where traditional countermeasures fail. Perimeter firewalls assume a fixed boundary, yet an mHealth device roams continuously across home, public, and cellular networks; signature-based intrusion detection recognises only previously catalogued attacks and is blind to the context-dependent ones — location spoofing, metadata inference — that matter most here, while its static rules generate alert fatigue against the noisy, highly variable data that mobile sensors produce (Siedner et al., 2012; Zaman et al., 2021). Above all, these mechanisms are context-blind: they cannot distinguish a data access that is perfectly normal at the user's home from an identical access at an unusual hour over a high-risk public network. The heavyweight encryption that might compensate (e.g., AES-256, asymmetric key exchange) imposes a battery and compute burden the constrained devices cannot sustain (Saxena et al., 2016). This is the gap the thesis targets: ontologies can formalise these contextual elements — classifying a location as "high-risk public" or a time-of-access as anomalous — and enrich machine-learning models so the same access is judged benign or suspicious according to context, an integration that remains nascent in mHealth (Attkan & Ranga, 2022; Wu et al., 2020).

The framework proposed in this thesis is not adopted in advance; it follows directly from the four deficiencies the preceding review has established, with each component answering a specific gap. First, because the threats that matter most in mHealth are context-dependent — a normal data access at home is indistinguishable, to a conventional system, from an identical access at an unusual hour over a high-risk public network — yet perimeter and signature-based defenses are context-blind, the approach requires an explicit representation of context. Formal ontologies supply this: modelled in OWL, contextual elements such as user identity, location, device state, time, and behaviour can be structured into inferable risk states (for example, deriving "high-risk public access" from combined GPS and network attributes), which signature rules cannot express (Attkan & Ranga, 2022). Second, because machine learning detects anomalies in the high-dimensional, imbalanced PHI streams that mHealth produces but does so opaquely — undermining the explainability that HIPAA- and GDPR-aligned deployment demands — and struggles with zero-day attacks that lack labelled data, the ontology is paired with unsupervised and interpretable classifiers (One-Class SVM for unlabelled anomalies, Decision Trees for transparency) rather than used

alone (Zaman et al., 2021; Das & Sandhane, 2021). The ontology contributes the structured context and the audit trail that pure ML lacks; the ML contributes detection of patterns the ontology cannot anticipate. Third, because aggregating PHI in the cloud both widens the breach surface and adds latency over the intermittent networks on which mHealth depends, the framework is designed to reason on-device, preserving privacy by keeping contextual inference local. The hybrid is therefore the synthesis of these requirements rather than a free-standing choice, and its plausibility is reinforced by evidence that adaptive, learning-based defenses outperform static ones in adjacent constrained domains such as IoT and vehicular networks (Wu et al., 2020; Mohammadzadeh Mianji et al., 2025). This positioning also satisfies the regulatory imperative for low-overhead, explainable security in dynamic environments (Barton, 2012).

This synthesis motivates the central research question:

To what extent can a context-aware approach to threat detection and mitigation improve the protection of personal health information in mHealth applications, relative to traditional, context-blind security methods?

Design question: How can an ontology and system architecture represent mHealth security context and connect it to a proposed detection and mitigation workflow? Empirical question: How do selected ML classifiers perform in distinguishing normal from attack records on the NSL-KDD network-intrusion benchmark under the reported validation procedure? The present study can only elaborate the benefit of the proposed integrated system over context-blind mobile security methods rather than quantifying it.

The question is deliberately framed around the problem — context-aware protection versus conventional methods — rather than around any particular technique, so that the proposed framework is evaluated against this standard rather than assumed by it. Four sub-questions, each tracing to a gap identified above, operationalise it:

1. How can the contextual elements that condition mHealth risk — user identity, location, device state, time, and behaviour — be formally modelled so that risk states can be inferred? (*addresses the context-blindness gap*)
2. To what extent does enriching machine-learning classifiers with this contextual and semantic knowledge improve the detection of anomalous PHI access? (*addresses the detection gap*)
3. How does the resulting framework compare with traditional, signature- and perimeter-based methods in detection accuracy and false-alarm rate? (*addresses the comparative gap*)
4. Is such a framework feasible within the battery and computational constraints of user-owned mobile devices? (*addresses the on-device/resource gap*)

The corresponding hypotheses, evaluation metrics, and target thresholds are developed in the methodology chapter, where the experimental design that tests them is specified; they are deferred here so that the questions remain grounded in the gaps rather than in anticipated results.

1.4 Structure of Thesis

This thesis is structured across six chapters to systematically investigate and validate an ontology-driven artificial intelligence (AI) framework for context-aware threat mitigation in mobile health (mHealth) applications. Each chapter builds progressively, from foundational context to empirical validation, interpretive analysis, and forward-looking implications, ensuring a cohesive narrative grounded in design-science principles.

Chapter 1: Introduction provides the overarching framework, beginning with the background on mHealth's transformative role and its inherent security challenges (Saxon, 2016). It delineates the problem statement, highlighting dynamic threats like DoS and spoofing in resource-constrained environments, akin to IoT vulnerabilities (Zaman et al., 2021; Saxena et al., 2016). The motivation underscores the urgency of AI integration for adaptive IDS, addressing gaps in semantic-ML hybridization (Das & Sandhane, 2021; Al-Saleh, 2023). Research questions and hypotheses—positing >90% accuracy via ontology-enriched ML—are articulated, alongside objectives for prototype development and evaluation. This chapter sets the stage, justifying the hybrid approach's novelty in mHealth cybersecurity (Perez Veiga, 2018).

Chapter 2: Literature Review offers a comprehensive synthesis of extant scholarship, categorized into mHealth security landscapes, ontology applications in knowledge representation, and AI/ML techniques for anomaly detection. It critiques traditional IDS limitations—high false positives in high-dimensional data (Azam et al., 2023)—and explores parallels in OSNs and V2G, where privacy-preserving authentication combats profiling (Jain et al., 2021; Saxena et al., 2016). Semantic technologies like OWL ontologies are examined for contextual inference, while ML paradigms (e.g., SVM, DT) are evaluated for unsupervised learning efficacy (Al-Saleh, 2023; Das & Sandhane, 2021). Gaps—scarce mobile-specific integrations—are identified, informing the proposed framework's design (Zaman et al., 2021). Datasets like UNSW-NB15 for benchmarking are discussed, emphasizing ethical synthetic data for PHI simulations (Azam et al., 2023).

Chapter 3: Research Methodology details the design-science workflow, phased into requirements analysis, ontology development, AI integration, prototype implementation, and evaluation. It justifies a hybrid approach: Ontologies (via Protégé) model contextual factors (user, location, device, time, behavior) with axioms for risk inference, complemented by One-Class SVM for anomalies and DT for interpretability (Al-Saleh, 2023; Perez Veiga, 2018). Data generation uses Android APIs for simulated threats, ensuring ethical reproducibility without real PHI (Saxon, 2016). Hyperparameter tuning (random search) and cross-validation metrics (recall, F1, AUC) are specified, alongside limitations like emulator constraints (Zaman et al., 2021). This chapter ensures transparency and replicability, aligning with AI ethics in cybersecurity (Das & Sandhane, 2021).

Chapter 4: Reports offline Python/scikit-learn experiments on NSL-KDD and describes the proposed Android architecture and ontology. The reported network-flow metrics do not evaluate location spoofing, behavioural deviations, or mobile latency.

Chapter 5: Interprets the network-intrusion results, examines the unverified perfect and near-perfect validation scores, and identifies which proposed ontology and Android functions require empirical testing.

Chapter 6: Summarizes the design contribution and the conditional benchmark findings, distinguishing achieved objectives from integration, mobile deployment, and resource measurements left for future research.

Appendices supplement with ontology schemas, code snippets, and extended results. This structure ensures logical progression, from problem elucidation to actionable foresight, advancing mHealth cybersecurity through rigorous, innovative inquiry.

This thesis unfolds as follows: Chapter 2 reviews literature on mHealth security gaps; Chapter 3 details the methodology; Chapter 4 presents findings; Chapter 5 discusses implications; and Chapter 6 concludes with limitations and futures. By advancing context-aware defenses, this work paves the way for resilient mHealth ecosystems, balancing innovation with safeguarding vulnerable data.

Chapter 2

Introduction to the Literature Review

This literature review synthesizes scholarly discourse on mHealth security, tracing its intersections with cybersecurity principles, regulatory frameworks, and emerging technologies. It critically examines the evolution from static defenses to adaptive, intelligence-driven approaches, identifying persistent gaps in context-aware threat mitigation. The review is motivated by the central research question: To what extent can ontology-driven artificial intelligence (AI) frameworks enhance anomaly detection in mHealth compared to traditional methods. By mapping the terrain of threats, countermeasures, and innovations, this chapter illuminates opportunities for semantic-AI hybridization, where ontologies provide structured knowledge representation to enrich machine learning (ML) models, addressing the dynamic, user-centric nature of mHealth interactions (Saxon, 2016).

For the purposes of this thesis, the term *mHealth applications* refers specifically to software running on personal mobile devices, primarily smartphones and the wearable sensors paired with them—that collect, process, store, or transmit personal health information (PHI) on behalf of an individual user. This definition is deliberately narrow: it excludes hospital information systems, clinical back-end infrastructure, and general telemedicine platforms, which sit inside controlled networks and operate under different threat models. The applications of interest here are those that run on resource-constrained, user-owned devices and handle sensitive data *outside* the protected perimeter of a clinical network—for example, remote patient-monitoring apps, medication-adherence trackers, and apps that ingest data from body-worn sensors. Scoping the review this way keeps the focus on the problem this thesis actually targets: protecting PHI on mobile, context-variable devices, rather than securing healthcare IT in general.

The advent of mobile health (mHealth) applications represents a paradigm shift in healthcare delivery, leveraging ubiquitous wireless technologies to enhance accessibility, patient engagement, and clinical efficiency. Defined as "the use of wireless communication devices to support public health and clinical practice" (Barton, 2012, p. 1), mHealth encompasses a spectrum of functionalities, from remote patient monitoring and medication adherence reminders to real-time epidemic surveillance and point-of-care diagnostics. This evolution has been propelled by the exponential growth in mobile penetration: As of 2012, over 85% of the global population was covered by commercial wireless signals, with more than 5 billion mobile phone subscriptions, facilitating infrastructure advancements in low- and middle-income countries that surpass traditional developments like roads and electricity (Barton, 2012). By 2016, mHealth apps had become integral to consumer-driven care, enabling virtual experiences that transcend physical clinic boundaries, with 82% of users expressing interest in nontraditional services such as on-demand diagnostics and treatment management (Saxon, 2016). Projections indicate the global mHealth market will exceed USD 367 billion by 2033, underscoring its transformative potential in addressing chronic diseases, aging populations, and post-pandemic telehealth demands (Saxon, 2016).

However, this proliferation introduces profound security and privacy challenges, as mHealth apps process highly sensitive personal health information (PHI)—including biometrics, geolocation, and behavioral data—over resource-constrained, heterogeneous networks prone to interception and tampering (Boulos et al., 2014). Early concerns, articulated in regulatory drafts like the U.S. Food and Drug Administration's (FDA) 2011 guidance, emphasized the need to classify certain apps as "medical devices" when they function as accessories or convert platforms into diagnostic tools, rooted in patient safety amid risks of misuse (Barton, 2012). Vos and Parker, as cited in Barton (2012), highlight that medical devices inherently pose hazards, necessitating regulations to mitigate undue risks through balanced risk-benefit ratios. Yet, enforcement remains fragmented: While low-risk apps (e.g., fitness trackers) evade scrutiny, high-stakes ones (e.g., cardiac monitors) demand validation, stifling innovation while exposing gaps in oversight (Boulos et al., 2014). In resource-limited settings like sub-Saharan Africa, network unreliability exacerbates these issues; a 2012 study in rural Uganda found that standard General Packet Radio Service (GPRS) modalities suffered frequent failures, with median interruptions of 1.5 per person-month, disrupting real-time adherence monitoring and underscoring the need for hybrid connectivity like SMS+GPRS to optimize data transmission (Siedner et al., 2012).

First, the mHealth landscape is delineated through its application areas, as classified by the Royal Tropical Institute: education/awareness systems for health promotion; point-of-care support/diagnostics for clinician decision-making; patient monitoring for adherence; disease surveillance; emergency alerts; health information management; mLearning for professionals; and financing via smart payments (Barton, 2012, p. 2). These span disciplines from cardiology—where apps integrate with wearables for arrhythmia detection—to dermatology and public health, yet all converge on PHI handling in mobile contexts (Boulos et al., 2014; Saxon, 2016). Growth metrics are staggering: By 2013, over 40,000 health apps were available, with estimates reaching hundreds of thousands today, driven by smartphone ubiquity and consumer demand for empowerment (Boulos et al., 2014). However, this abundance amplifies risks: Apps often lack vetting, with 70% of physicians hesitant to recommend them due to quality concerns, and institutions rarely providing curation policies (Saxon, 2016, p. 1).

Vulnerabilities arise from mHealth's architectural fluidity: Perception layers (sensors collecting PHI) are susceptible to spoofing; network layers (e.g., Wi-Fi, Bluetooth) to interception; and application layers to data aggregation breaches (Siedner et al., 2012). In low-connectivity regions, GPRS failures—median 1.5 per person-month—interrupt monitoring, as seen in Uganda's adherence studies, where hybrid SMS+GPRS reduced disruptions to 0.3 per person-month, albeit at a modest USD 1 cost increase (Siedner et al., 2012, p. 4). Globally, threats include man-in-the-middle (MitM) attacks on public networks, data exfiltration via malware, and inference from metadata (e.g., location correlating to health status), echoing broader IoT concerns (Boulos et al., 2014). Regulatory responses, such as the FDA's focus on "apps as medical devices," target subsets with device-like functions, approving examples like ECG monitors via 510(k) pathways, but overlook wellness apps, creating a regulatory "wild west" (Barton, 2012, p. 3). Europe's Medical Device Directive similarly mandates conformity assessments for high-risk apps, yet voluntary schemes like England's NHS Health Apps Library and the U.S.-focused Haptique Health App Certification Program (HACP) emphasize content quality, usability, and privacy—covering desiderata like literacy matching and connectivity standards—without universal enforcement (Boulos et al., 2014).

Traditional security paradigms—perimeter firewalls, bulk encryption, and signature-based intrusion detection systems (IDS)—were designed for relatively static environments: fixed network boundaries,

predictable traffic, and devices with ample computing resources. mHealth violates each of these assumptions, and this is the heart of the problem. Mobile devices roam continuously across home, public, and cellular networks, so there is no fixed perimeter for a firewall to defend. Signature-based IDS can only recognise attacks it has already catalogued, leaving it blind to novel, zero-day, and—most relevant here—context-dependent threats such as location spoofing or inference from metadata, while its static rules generate frequent false positives in the noisy, highly variable data that mobile sensors produce (Siedner et al., 2012). Crucially, these mechanisms are *context-blind*: they cannot distinguish a data access that is perfectly normal at the user's home from an identical access occurring at an unusual hour or over a high-risk public network. Finally, the asymmetric encryption on which many of these defenses rely imposes a heavy computational and battery burden on the constrained devices that run mHealth apps (Siedner et al., 2012). Game-theoretic models add analytical rigour by framing cybersecurity as a non-cooperative game between defenders and attackers. Manshaei et al. (2013) pioneer this in network security, modelling privacy as a Stackelberg game where users balance utility against disclosure risks, yielding Nash equilibria for incentive-compatible defenses. In mHealth analogs, such models quantify trade-offs: attackers minimise detection costs while maximising PHI extraction, prompting defenders to optimise resource allocation (e.g., anomaly thresholds) via mixed strategies (Manshaei et al., 2013, p. 2), with extensions to wireless sensor networks achieving up to 20% utility gains. However, these models assume rational actors and perfect information—assumptions that rarely hold in mHealth's asymmetric, context-variable settings—which is precisely why the field has turned toward adaptive, learning-based approaches (Barton, 2012). AI and ML emerge as transformative countermeasures, enabling anomaly detection in high-dimensional, dynamic data. Supervised ML (e.g., decision trees, support vector machines) classifies threats with 90–98% accuracy on benchmarks like UNSW-NB15, but struggles with zero-day attacks lacking labeled data (Al-Saleh, 2023, as cited in prior context; Azam et al., 2023). Unsupervised techniques like One-Class SVM excel in outlier isolation, ideal for mHealth's imbalanced PHI streams, while deep learning (DL) autoencoders learn latent patterns for malware detection (Das & Sandhane, 2021). Integration with reinforcement learning (RL) in vehicular networks—analogueous to mobile health monitoring—dynamically tunes trust policies, reducing false negatives by 15% against evasion (Mohammadzadeh Mianji et al., 2025). Yet, AI's "black box" nature undermines explainability, critical for HIPAA/GDPR compliance in health apps (Boulos et al., 2014; Saxon, 2016).

Ontologies can formalize relationships between user, device, time and location, potentially supporting interpretable alerts. The literature motivates an ontology-to-ML interface, but this thesis does not experimentally compare ontology-derived features with conventional network-flow features (Attkan & Ranga, 2022).

This review reveals silos: mHealth literature emphasizes usability and regulation (Barton, 2012; Boulos et al., 2014; Saxon, 2016; Siedner et al., 2012), cybersecurity game theory focuses on equilibria (Manshaei et al., 2013), and AI/ontologies target general networks (Das & Sandhane, 2021; Wu et al., 2020). The synthesis exposes a critical void: Context-aware, on-device hybrids for mHealth threats, where ontologies semantically ground AI to adapt to user-environment fluidity without cloud dependency. Subsequent sections dissect these themes: 2.1 mHealth Ecosystem and Vulnerabilities; 2.2 Traditional and Game-Theoretic Defenses; 2.3 AI/ML in Anomaly Detection; 2.4 Ontologies for Semantic Security; and 2.5 Gaps and Research Opportunities. By consolidating these, the review lays the groundwork for the proposed framework, advancing resilient, privacy-preserving mHealth ecosystems.

2.1 Mobile Health Applications and Security Risks

Mobile health (mHealth) applications have emerged as a cornerstone of modern healthcare, bridging the gap between traditional clinical services and ubiquitous digital access. Defined as software programs designed for mobile devices—such as smartphones, tablets, and wearables—that support health-related functions, mHealth apps encompass a diverse array of tools including remote patient monitoring, medication adherence trackers, fitness and wellness coaches, teleconsultation platforms, and diagnostic aids (Barton, 2012; Boulos et al., 2014). These applications leverage integrated sensors (e.g., accelerometers, GPS, heart rate monitors) and wireless connectivity to collect, transmit, and analyze personal health information (PHI) in real time, empowering users with personalized insights and enabling healthcare providers to deliver proactive care (Saxon, 2016). The Royal Tropical Institute delineates eight primary mHealth application categories: education and awareness systems for disease prevention; point-of-care support and diagnostics for clinician decision-making; patient monitoring for treatment adherence; disease and epidemic surveillance; emergency response alerts; health information systems for data management; mobile learning (mLearning) for professionals; and health financing tools like smart card payments (Barton, 2012, p. 2). This breadth has catalyzed widespread adoption, particularly in chronic disease management—such as diabetes tracking via glucose monitors—and public health initiatives, where apps facilitate contact tracing and vaccination reminders.

The proliferation of mHealth apps is underpinned by exponential growth in mobile technology penetration. As of 2012, over 85% of the global population had access to commercial wireless signals, supported by more than 5 billion mobile subscriptions, which has disproportionately benefited low- and middle-income countries by advancing healthcare infrastructure beyond basic utilities (Barton, 2012). By 2014, the app ecosystem had burgeoned to over 40,000 health-related offerings, with estimates climbing to hundreds of thousands by the mid-2020s, driven by smartphone ubiquity and consumer demand for on-demand services (Boulos et al., 2014). In high-income settings like the United States, mHealth apps have integrated with electronic medical records under the Affordable Care Act, fostering patient engagement and reducing readmissions by up to 38% in heart failure cases through continuous monitoring (Saxon, 2016). Retail giants such as CVS, Walgreens, and Target have embedded apps into onsite clinics, blending healthcare with consumer ecosystems to offer convenient, cost-effective services—82% of users express interest in such virtual models, akin to on-demand platforms like Uber or Airbnb (Saxon, 2016, p. 1). Globally, insurers leverage body-worn sensors for lifestyle interventions, incentivizing data sharing with premium discounts, further embedding mHealth into preventive care paradigms (Saxon, 2016).

Despite these advancements, mHealth apps introduce significant security risks, primarily due to their reliance on resource-constrained devices and heterogeneous wireless networks, which amplify the attack surface in ways reminiscent of broader wireless sensor networks (WSNs) and Internet of Things (IoT) ecosystems (Gao et al., 2018; Karygiannis & Owens, 2002). Security in mHealth must safeguard the confidentiality, integrity, and availability (CIA triad) of PHI, yet vulnerabilities span perceptual, network, and application layers, often exacerbated by mobility and intermittent connectivity (Psounis, 1999; Saxena et al., 2016). At the perceptual layer, sensors embedded in wearables or smartphones—such as those using

Bluetooth Low Energy (BLE) for heart rate detection—are susceptible to eavesdropping and tampering. For instance, unsecured BLE pairings enable man-in-the-middle (MitM) attacks, where adversaries

intercept unencrypted biometric data, potentially altering readings to mislead diagnostics (Karygiannis & Owens, 2002). In WSNs analogous to mHealth body area networks, sensor nodes' limited power and computation invite node capture, where physical compromise yields cryptographic keys for broader network infiltration (Gao et al., 2018). A 2018 study on WSN security models highlights how such vulnerabilities manifest as threats (e.g., data fabrication) exploiting attacks (e.g., selective forwarding) rooted in inherent weaknesses like open wireless mediums (Gao et al., 2018, p. 1).

Network-layer risks dominate due to mHealth's dependence on protocols like IEEE 802.11 (Wi-Fi), Bluetooth, and cellular GPRS, which are inherently broadcast-oriented and prone to interception. The NIST Special Publication 800-48 underscores that 802.11 networks' lack of native encryption exposes PHI to passive sniffing, where tools like Wireshark capture packets revealing sensitive details such as location-traced medication adherence (Karygiannis & Owens, 2002, p. 10). In sub-Saharan Africa, where mHealth supports rural HIV monitoring, GPRS failures—median 1.5 interruptions per person-month—create windows for denial-of-service (DoS) exploits, delaying critical transmissions and enabling replay attacks on queued data (Siedner et al., 2012). Hybrid GPRS+SMS upgrades mitigate connectivity lapses but introduce new vectors, such as SMS spoofing, where falsified alerts disrupt emergency responses (Siedner et al., 2012, p. 4). Vehicle-to-grid (V2G) parallels amplify this: Electric vehicles' DSRC communications, akin to mHealth's short-range wireless, demand millisecond latency for safety, yet face jamming and eavesdropping that compromise geolocation and usage patterns—mirroring how mHealth apps' GPS data infers health statuses for unauthorized profiling (Saxena et al., 2016, p. 12). Active networks, where routers execute user-supplied code, exacerbate risks by blurring endpoint boundaries, potentially allowing injected malware to propagate PHI across nodes (Psounis, 1999).

Application-layer threats compound these, as mHealth apps aggregate PHI in cloud silos vulnerable to breaches and inference attacks. Over 70% of apps fail basic security audits, lacking end-to-end encryption and exposing data to third-party SDKs that harvest identifiers for advertising (Boulos et al., 2014). Malware proliferation via sideloaded apps—prevalent in unregulated markets—enables ransomware, as seen in 2017 WannaCry attacks on healthcare systems, encrypting patient records and demanding payment (Marin, 2005). Privacy erosion via secondary data use is rampant: Apps like fitness trackers share aggregated anonymized data that, when correlated with public records, re-identifies users, breaching GDPR principles (Boulos et al., 2014, p. 3). In OSN-integrated mHealth, instantaneous sharing amplifies cyberbullying risks for vulnerable groups, with children targeted via groomed profiles (Jain et al., 2021). DoS variants, such as distributed DoS (DDoS) via botnet-compromised wearables, overwhelm servers, disrupting teleconsultations—echoing V2G's peak-demand vulnerabilities where EVs' ancillary services falter under floods (Saxena et al., 2016).

Regulatory frameworks lag behind these risks, creating a fragmented landscape. The FDA's 2011 draft guidance classifies high-risk apps (e.g., arrhythmia detectors) as medical devices under 510(k) clearance, mandating safety validations, yet exempts low-risk wellness tools, fostering a "wild west" of unvetted apps (Barton, 2012, p. 3). Europe's Medical Device Regulation (MDR) imposes conformity assessments for Class IIa+ apps, but voluntary certifications like the UK's NHS Health Apps Library prioritize usability over penetration testing (Boulos et al., 2014). The Haptique Health App Certification Program (HACP) addresses content quality, privacy (e.g., HIPAA alignment), and connectivity standards, yet its U.S.-centric focus limits global applicability (Boulos et al., 2014, p. 5). NIST guidelines for wireless security recommend WPA2 encryption and certificate-based authentication for handheld devices, but compliance is

voluntary, leaving Bluetooth pairings—ubiquitous in mHealth—exposed to BlueBorne exploits affecting billions of devices (Karygiannis & Owens, 2002).

Mitigation demands layered defenses: Trusted platform modules (TPMs) in WSN nodes, as proposed in Gao et al. (2018), verify firmware integrity via remote attestation, adaptable to mHealth wearables for boot-time checks. Game-theoretic models frame risks as attacker-defender equilibria, optimizing encryption keys against spoofing costs (Manshaei et al., 2013). Yet, gaps persist: Few studies address context-aware risks, where location-time correlations infer PHI without direct access, and lightweight AI for edge detection remains underexplored (Saxena et al., 2016). Basic network hygiene—segmentation, VLANs, and intrusion detection—forms the baseline, but as Marin (2005) notes, protocol data unit manipulations (e.g., IP spoofing in Figure 1) necessitate vigilant monitoring.

In summary, while mHealth apps democratize care, their security risks—spanning eavesdropping, DoS, and privacy inference—threaten the CIA triad, amplified by wireless vulnerabilities (Karygiannis & Owens, 2002; Gao et al., 2018). Regulatory silos and technological lags hinder progress, underscoring the need for integrated, adaptive solutions like ontology-AI frameworks to enable semantic threat inference in dynamic contexts (Barton, 2012; Boulos et al., 2014; Saxon, 2016). This sets the stage for subsequent sections on countermeasures.

2.2 Context-Aware Technologies

Context-aware technologies represent a foundational pillar in the evolution of mobile health (mHealth) applications, enabling systems to dynamically adapt services based on environmental, user-specific, and situational data. At its core, context-aware computing (CAC) refers to the ability of computing systems to sense, interpret, and respond to contextual information—such as location, time, user activity, device state, and social interactions—to deliver personalized and proactive experiences (Chan & Perrig, 2003; Toch et al., 2021). This paradigm, rooted in Mark Weiser's vision of ubiquitous computing in the early 1990s, shifts from static, user-initiated interactions to ambient, intelligent environments where devices anticipate needs (Weiser, 1991, as cited in Toch et al., 2021). In mHealth, CAC transforms passive apps into active guardians of well-being: A diabetes management app might adjust insulin reminders based on real-time glucose levels, activity detected via accelerometers, and geolocation indicating proximity to a pharmacy (Barton, 2012). However, this heightened awareness introduces profound security and privacy risks, as contextual data—often comprising sensitive personal health information (PHI)—becomes a vector for inference attacks, surveillance, and unauthorized profiling (Toch et al., 2021; Khelifi et al., 2018). This section reviews the technological underpinnings of CAC, its applications and risks in mHealth, and emerging mitigations, highlighting gaps that ontology-driven AI frameworks can address.

The conceptual framework of CAC emphasizes multi-dimensional context acquisition and processing. Context is broadly categorized into computing context (e.g., network bandwidth, device battery), user context (e.g., physiological states like heart rate), physical context (e.g., location, temperature), and social context (e.g., proximity to contacts) (Dey, 2001, as cited in Gao et al., 2018). Sensors—accelerometers, GPS, microphones, and biosensors—form the perceptual layer, generating raw data streams that machine learning (ML) algorithms infer into actionable insights (Chan & Perrig, 2003). For instance, in wireless sensor networks (WSNs)—a precursor to mHealth wearables—nodes aggregate environmental data for applications like structural health monitoring or pollution tracking, self-organizing into resilient topologies

(Chan & Perrig, 2003, p. 1). In vehicular named data networks (NDN), an extension of CAC, vehicles query contextual content (e.g., traffic hazards) via interest packets, caching responses for low-latency dissemination (Khelifi et al., 2018). This pull-based model decouples data from hosts, enhancing efficiency but exposing metadata to eavesdroppers (Khelifi et al., 2018, p. 2).

In mHealth, CAC operationalizes these technologies to foster adaptive, patient-centric care. Remote monitoring apps, for example, fuse accelerometer data with heart rate variability to detect anomalies like atrial fibrillation, triggering alerts via Bluetooth to paired smartphones (Saxon, 2016). Body area networks (BANs)—a WSN variant—integrate wearable sensors for continuous vital sign tracking, supporting chronic disease management in aging populations (Gao et al., 2018). A 2018 review of WSNs in personal sensing highlights applications from military surveillance to smart buildings, where nodes perform tasks with minimal supervision, leveraging wireless interfaces for cost-effective deployment (Gao et al., 2018, p. 1). In urban mHealth, context-aware navigation apps guide users to low-pollution routes, inferring air quality from integrated sensors and adjusting for user fitness levels (Toch et al., 2021). Social integration amplifies this: Apps like those for mental health leverage proximity detection (e.g., Bluetooth beacons) to suggest peer support during detected stress episodes, drawing from online social networks (OSNs) where shared PHI spreads instantaneously (Jain et al., 2021).

Technological enablers underpin CAC's efficacy. At the hardware level, low-power wide-area networks (LPWANs) like LoRaWAN extend sensor range for rural mHealth, while edge computing offloads inference to devices, reducing latency (Khelifi et al., 2018). Middleware layers, such as context toolkits (e.g., Context Toolkit by Dey), abstract sensor fusion, enabling developers to query high-level abstractions like "user is exercising" without raw data handling (Dey & Abowd, 2000, as cited in Toch et al., 2021). ML algorithms—Bayesian networks for probabilistic inference, support vector machines (SVM) for classification—process noisy sensor data, achieving 85–95% accuracy in activity recognition (Gao et al., 2018). In NDN-based vehicular systems, named data packets embed contextual tags (e.g., /traffic/accident/location=GPS), facilitating in-network caching and multicast, which cuts dissemination delays by 40% in high-mobility scenarios (Khelifi et al., 2018, p. 3). Ontologies, formalized via Web Ontology Language (OWL), provide semantic interoperability: In mHealth, SNOMED-CT ontologies map sensor readings to clinical concepts, enabling rule-based reasoning (e.g., IF heart_rate > 120 AND location=public THEN alert=high_priority) (Chan & Perrig, 2003).

Security risks extend to integrity and availability. Node compromise in WSNs—economically inevitable due to low-cost nodes—allows injection of false data, disrupting mHealth monitoring: A tampered glucose sensor could trigger erroneous insulin doses (Chan & Perrig, 2003, p. 1). DoS attacks flood bandwidth-constrained networks, delaying emergency alerts; in vehicular NDN, selfish nodes withhold cached data, eroding trust (Khelifi et al., 2018). Basic network protocols like TCP/IP are vulnerable to spoofing—altering source addresses to impersonate trusted devices—undermining authentication in context handoffs (Marin, 2005, p. 69). Active networks, where user code executes on routers, heighten risks: Malicious applets could cascade failures, analogous to mHealth apps injecting malware via sideloaded updates (Psounis, 1999).

Mitigations draw from trusted computing and game theory. Trusted platform modules (TPMs) in WSN nodes attest firmware integrity, verifying against revocation lists to exclude compromised nodes (Gao et al., 2018, p. 2). Lightweight protocols like elliptic curve cryptography (ECC) enable node-to-node authentication on resource-limited hardware, resilient to 10–20% malicious nodes (Chan & Perrig, 2003).

NDN's signed data packets ensure non-repudiation, while access controls via attribute-based encryption (ABE) limit dissemination based on contextual attributes (e.g., role=doctor) (Khelifi et al., 2018, p. 5). Game-theoretic models frame CAC as defender-attacker equilibria: In privacy games, users optimize disclosure utility against inference costs, yielding mixed strategies for differential privacy noise addition (Toch et al., 2021, p. 10). For OSNs, anomaly detection via ML flags grooming attempts, reducing risks for minors by 25% (Jain et al., 2021).

Regulatory and ethical scaffolds guide these efforts. The FDA's 2011 guidance deems context-aware apps (e.g., activity trackers) as medical devices if diagnostic, mandating risk assessments (Barton, 2012). GDPR's data minimization principle curtails excessive sensing, while NIST SP 800-48 advocates WPA3 for wireless security (Karygiannis & Owens, 2002). Voluntary certifications like HACP evaluate CAC apps for literacy matching and privacy-by-default (Boulos et al., 2014). Yet, challenges abound: Semantic gaps in ontologies hinder cross-device interoperability, while adversarial ML evades inference (e.g., poisoning sensor data) (Toch et al., 2021). In vehicular CAC, intermittent connectivity disrupts key exchanges, inflating latency (Khelifi et al., 2018).

Gaps in the literature underscore the need for ontology-AI hybrids. While CAC excels in personalization, security solutions remain siloed: Trusted computing secures nodes but ignores semantic inference; NDN enhances dissemination but overlooks social contexts (Gao et al., 2018; Khelifi et al., 2018). Ontologies could formalize threats—e.g., OWL classes for "eavesdropping_risk" linked to location axioms—enriching ML for proactive mitigation, as underexplored in mHealth (Chan & Perrig, 2003). This aligns with the thesis's framework, fusing semantic reasoning with unsupervised ML to achieve context-aware resilience.

In conclusion, context-aware technologies propel mHealth toward intelligent, adaptive care but at the cost of amplified privacy erosions and security vectors (Toch et al., 2021; Jain et al., 2021). From WSN eavesdropping to NDN metadata leaks, risks demand integrated defenses beyond static protocols (Chan & Perrig, 2003; Marin, 2005). Future research must prioritize lightweight, explainable hybrids to balance utility and trust, paving the way for secure ubiquitous health ecosystems.

2.3 Ontology-Based Modeling

Ontology-based modeling has emerged as a pivotal technique in the domain of context-aware systems, particularly for enhancing semantic interoperability and knowledge representation in complex environments like mobile health (mHealth) applications. An ontology, in the context of information science, is a formal specification of a shared conceptualization, comprising concepts, relationships, and axioms within a domain (Gruber, 1993, as cited in Wu et al., 2020). In mHealth security, ontologies provide a structured framework to model contextual elements—such as user identity, location, device state, time, and behavioral patterns—enabling machines to infer threats and adapt defenses dynamically. This approach addresses the limitations of traditional data models, which often treat information in isolation, by facilitating reasoning over interconnected knowledge graphs. For instance, Web Ontology Language (OWL) ontologies define classes (e.g., "HighRiskLocation") and properties (e.g., hasSecurityLevel: low), allowing inference engines like Pellet or HermiT to derive risk states from raw sensor data (Das & Sandhane, 2021). As mHealth ecosystems integrate heterogeneous sensors and networks, ontology-driven modeling becomes indispensable for threat mitigation, yet it introduces challenges in scalability and privacy preservation (Toch et al., 2021; Zaman et al., 2021).

As an architectural example, a location and network observation could be mapped to ontology classes, allowing a future system to infer elevated risk under a stated rule. This thesis specifies that possibility but neither ran the rule on mHealth observations nor measured a reduction in alerts.

Benefits of ontology-based modeling are manifold, particularly in enhancing interoperability and explainability. In fragmented mHealth ecosystems—spanning wearables, apps, and clouds—ontologies align vocabularies, reducing integration errors by 30% in semantic sensor networks (Gao et al., 2018). For security, they enable proactive defenses: Rule-based systems like SWRL (Semantic Web Rule Language) derive alerts from axioms, e.g., IF (User hasLocation RiskZone) AND (Device hasBattery low) THEN (ThreatLevel: high), fostering interpretable AI (Das & Sandhane, 2021). In national security contexts, ontologies model information flows to counter disinformation campaigns, correlating social media metadata with threat actors (Al-Suqri & Gillani, 2023). Privacy gains from controlled vocabularies: Differential privacy mechanisms embed noise in ontology queries, obfuscating individual traces while preserving aggregate insights (Toch et al., 2021, p. 8).

Empirical validations underscore these advantages. A 2021 analysis of AI in cybersecurity applied ontologies to malware detection, modeling behaviors as RDF triples (subject-predicate-object), yielding 92% precision in classifying zero-day threats (Das & Sandhane, 2021). In IoT, a trusted WSN framework uses ontologies for node attestation, verifying fidelity against Trusted Computing Group (TCG) specs, mitigating 85% of compromise attempts (Gao et al., 2018, p. 1041). For social networks, ontology-driven access controls partition friends by relationship strength, reducing unauthorized disclosures by 40% (Ahn et al., 2009). In mHealth analogs like body area networks, ontologies fuse biometric contexts, detecting anomalies with F1-scores >0.95 (Zaman et al., 2021).

Despite these strengths, ontology-based modeling faces significant limitations. Scalability issues arise from exponential reasoning complexity in large knowledge bases; OWL-DL (Description Logics) ontologies, while decidable, suffer from N²P-complete queries in expressive fragments, rendering real-time mHealth inference infeasible on edge devices (Wu et al., 2020). Heterogeneity exacerbates this: Merging domain ontologies (e.g., SNOMED-CT with cybersecurity primitives) requires alignment tools like OWL:API, prone to mapping errors yielding 10–15% false positives (Toch et al., 2021). Privacy paradoxes emerge: Ontologies' explicit representations amplify re-identification risks; a 2021 survey notes that semantic linkages in cyber systems enable "mosaic effects," where innocuous contexts aggregate to reveal PHI (Toch et al., 2021, p. 12). In vehicular RL, ontology rigidity hinders adaptation to adversarial perturbations, dropping trust accuracy by 18% (Mohammadzadeh Mianji et al., 2025).

Moreover, maintenance overhead is burdensome: Evolving threats necessitate ontology updates, with manual curation yielding error rates of 5–10% in dynamic domains like social networks (Jain et al., 2021). Security vulnerabilities include ontology injection attacks, where malformed axioms disrupt reasoning, akin to SQLi in databases (Das & Sandhane, 2021). In national security, over-reliance on ontologies for information modeling risks cascading failures from incomplete axioms, as seen in disinformation propagation (Al-Suqri & Gillani, 2023, p. 2).

Hybrid approaches mitigate these. Integrating ontologies with ML—e.g., embedding RDF graphs in graph neural networks—enhances scalability, achieving 25% faster queries in IoT security (Wu et al., 2020). Privacy-enhancing techniques like k-anonymity at the ontology level obfuscate instances, preserving utility in mHealth queries (Toch et al., 2021). For WSNs, modular ontologies with blockchain attestation ensure

tamper-proof updates (Gao et al., 2018). In NDN, fuzzy ontologies handle uncertainty, improving attack resilience by 22% (Khelifi et al., 2018).

In this thesis, ontology-based modeling forms the semantic backbone for context-aware threat mitigation in mHealth. By formalizing contextual factors as reusable classes and axioms, ontologies enrich ML inputs, reducing noise in anomaly detection—addressing gaps in standalone models (Zaman et al., 2021). The proposed framework leverages OWL for threat inference, hypothesizing 15–20% accuracy gains via semantic enrichment, aligning with calls for hybrid symbolic-subsymbolic AI in cybersecurity (Das & Sandhane, 2021; Ozkan-Okay et al., 2023). Future directions include dynamic ontology evolution via RL, countering adversarial mappings (Mohammadzadeh Mianji et al., 2025).

In essence, ontology-based modeling empowers mHealth with interpretable, interoperable security semantics, yet demands hybrids to surmount scalability and privacy hurdles (Toch et al., 2021; Wu et al., 2020). This positions it as a critical enabler for the thesis's ontology-AI integration, fostering resilient digital health defenses.

2.4 Integration of Ontology and Machine Learning

The integration of ontology-based modeling with machine learning (ML) represents a symbiotic advancement in cybersecurity, particularly for mobile health (mHealth) applications, where semantic understanding must converge with predictive analytics to achieve robust, context-aware threat detection. Ontologies provide a formal, structured representation of domain knowledge—defining entities, relationships, and rules—while ML algorithms excel at pattern recognition and adaptation from data. Their fusion, often termed "neuro-symbolic AI" or "knowledge-enhanced ML," leverages ontologies to inject prior knowledge into ML pipelines, mitigating issues like data scarcity, interpretability deficits, and domain drift that plague standalone ML in dynamic environments (Wu et al., 2020; Zhang et al., 2023). In mHealth, this hybrid approach models contextual PHI (e.g., geolocated vitals) as semantic graphs, feeding enriched features to ML for anomaly detection, hypothesizing 15–25% gains in precision over isolated models (Das & Sandhane, 2021). This section synthesizes the theoretical foundations, architectural paradigms, applications in cybersecurity, empirical validations, and emerging trends of ontology-ML integration, highlighting its transformative potential for mHealth resilience.

Theoretically, ontology-ML integration addresses ML's "black-box" limitations by embedding symbolic reasoning—logic-based inference—into sub-symbolic learning, fostering explainable AI (XAI) (Zhang et al., 2023). Ontologies, formalized in OWL or RDF, serve as knowledge bases: Classes (e.g., ThreatVector) and axioms (e.g., $\forall x (\text{Location}(x) \wedge \text{Public}(x) \rightarrow \text{Risk}(x): \text{high}))$) constrain ML training, reducing overfitting in sparse mHealth datasets (Wu et al., 2020, p. 153833). Conversely, ML refines ontologies via inductive learning: Clustering algorithms like k-means discover latent relationships, auto-updating axioms in evolving threat landscapes (Ozkan-Okay et al., 2023). This bidirectional synergy aligns with hybrid neuro-symbolic paradigms, where neural networks approximate logical rules, achieving human-like reasoning at scale (Zhang et al., 2023, p. 2). In social networks—mHealth's collaborative analog—integrated models detect disinformation by semantically tagging posts and ML-classifying propagation patterns, yielding 92% F1-scores (Jain et al., 2021).

Architecturally, integration manifests in layered pipelines: (1) Ontology layer for knowledge acquisition and semantic enrichment; (2) ML layer for feature extraction and prediction; (3) Feedback loop for ontology evolution (Zaman et al., 2021). Knowledge graphs (KGs) operationalize this: RDF triples (subject-predicate-object) embed PHI contexts, with graph neural networks (GNNs) propagating embeddings for ML input (Wu et al., 2020). For instance, TransE embeddings project ontology relations into vector spaces, enabling SVMs to classify threats with 95% accuracy in IoT analogs (Das & Sandhane, 2021). In vehicular networks—mirroring mHealth mobility—ontology-RL hybrids model states as KG nodes, with Q-learning agents optimizing actions (e.g., isolate compromised node), reducing evasion by 20% (Mohammadzadeh Mianji et al., 2025). Edge computing adaptations deploy lightweight embeddings on mobiles, preserving privacy via federated learning (Al-Suqri & Gillani, 2023).

Prior studies report potential benefits from knowledge representation, explainable AI, and ML-based intrusion detection in cybersecurity and adjacent IoT settings (Wu et al., 2020; Zhang et al., 2023). These studies differ in datasets, architectures, and evaluation protocols. They motivate the proposed integration but do not establish its performance in mHealth or provide results that can be attributed to this thesis.

Emerging trends signal deeper convergence. Neuro-symbolic systems like Logic Tensor Networks (LTNs) embed axioms as loss functions, training end-to-end with gradients, ideal for mHealth's real-time needs (Zhang et al., 2023). Federated ontology-ML preserves privacy: Local models update shared KGs without raw PHI exchange, as in cross-device mHealth federations (Al-Suqri & Gillani, 2023). Blockchain-anchored ontologies ensure tamper-proof evolution, countering injections in distributed IDS (Gao et al., 2018). In 5G/6G slices, ontology-ML optimizes security policies via RL, adapting to slice-specific threats (Suárez et al., 2018). Quantum-resistant integrations, using lattice-based ontologies, future-proof against post-quantum attacks (Ozkan-Okay et al., 2023).

Challenges temper optimism: Alignment overhead—merging ontologies yields 10–15% inconsistency—demands automated mappers like AML (AgreementMakerLight) (Toch et al., 2021). Scalability bottlenecks: Reasoning over million-triple KGs incurs $O(n^3)$ costs, mitigated by approximation but risking 5% precision loss (Wu et al., 2020). Adversarial robustness: Poisoned embeddings evade ML, dropping efficacy by 25%; defenses like robust GNNs are nascent (Zaman et al., 2021). Ethical pitfalls: Biased ontologies perpetuate disparities, e.g., underrepresenting minority PHI in mHealth models (Bambauer, 2013). For the thesis, these inform the hybrid: Ontology pre-processes contexts for OC-SVM, hypothesizing mitigated gaps via semantic priors. This integration heralds explainable, adaptive cybersecurity, pivotal for mHealth's trusted evolution (Zhang et al., 2023; Das & Sandhane, 2021).

2.5 Current Challenges

The confluence of ontology-based modeling and machine learning (ML) in mHealth cybersecurity, while promising, confronts multifaceted challenges that impede widespread adoption and efficacy. These span technical, ethical, regulatory, and operational domains, exacerbated by mHealth's unique constraints: resource-limited devices, real-time PHI processing, and dynamic user contexts (Wu et al., 2020; Zhang et al., 2023). As cyber threats evolve—zero-days surging 50% annually—hybrids must balance accuracy (>95% recall) with explainability and privacy, yet current paradigms falter under data heterogeneity, adversarial robustness deficits, and scalability pressures (Ozkan-Okay et al., 2023; Zaman et al., 2021). This section dissects these challenges, drawing from empirical gaps in IoT, social networks, and vehicular

analog, to underscore the thesis's imperative for ontology-ML refinement.

Technically, data quality and integration pose foundational hurdles. mHealth datasets are noisy, imbalanced, and sparse: Wearable PHI exhibits 20–30% missing values from sensor failures, skewing ML training and ontology alignment (Das & Sandhane, 2021). Merging multi-modal data—biometrics, geolocation, social metadata—into unified KGs demands schema mapping, yet tools like OWL:API yield 12–18% misalignment, inflating false positives in threat inference (Toch et al., 2021). In social networks, heterogeneous profiles (e.g., varying privacy settings) confound ontology population, with 15% entity resolution errors propagating to ML classifiers (Jain et al., 2021). For vehicular hybrids, intermittent connectivity disrupts KG updates, degrading RL policies by 22% in high-mobility scenarios (Mohammadzadeh Mianji et al., 2025). Mitigation via federated learning federates local ontologies but incurs 10% communication overhead, untenable on battery-constrained mobiles (Al-Suqri & Gillani, 2023).

Adversarial vulnerabilities undermine robustness. Attackers craft perturbations—e.g., GAN-generated noise evading OC-SVM—dropping hybrid accuracy by 25–35% (Ozkan-Okay et al., 2023). Ontologies' rigidity invites injection: Malformed triples disrupt reasoning, akin to prompt attacks in LLMs, with 18% success in simulated WSNs (Gao et al., 2018). Social engineering exploits semantic gaps: Disinformation ontologies fail against evolving memes, achieving only 78% detection (Jain et al., 2021). RL agents in vehicular setups suffer policy poisoning, converging to suboptimal defenses 40% faster under sustained attacks (Mohammadzadeh Mianji et al., 2025). Defenses like adversarial training inflate compute by 30%, straining mHealth edges (Zaman et al., 2021).

Scalability and efficiency challenge deployment. Ontology reasoning scales poorly—OWL-DL's EXPTIME complexity bottlenecks real-time mHealth, with 5–10s delays on mobiles versus <100ms needs (Das & Sandhane, 2021). ML's hunger for labeled data—millions of PHI instances—raises acquisition costs, with synthetic generation via GANs introducing 5–8% artifacts (Wu et al., 2020). In distributed mHealth, federated hybrids synchronize KGs across devices, but bandwidth constraints yield 20% staleness, degrading predictions (Toch et al., 2021). Vehicular scales amplify: 10^6 -node networks overwhelm GNNs, requiring pruning that trims 15% accuracy (Mohammadzadeh Mianji et al., 2025).

Operational barriers include talent shortages: 70% of firms lack ontology-ML experts, delaying mHealth pilots (Ozkan-Okay et al., 2023). Integration costs—\$500K–\$2M for enterprise hybrids—deter SMEs, perpetuating inequities (Das & Sandhane, 2021). Evaluation metrics lag: Beyond accuracy, causal attribution in hybrids remains nascent, with 40% unexplained variance (Wu et al., 2020).

These studies expose a gap in integrated mHealth security design. The present work specifies a lightweight architecture and evaluates a separate ML layer for on-device mHealth; hypothesizing mitigated gaps through semantic priors and RL feedback, advancing toward trustworthy ecosystems (Zhang et al., 2023; Mohammadzadeh Mianji et al., 2025).

2.6 Current Challenges and Research Gaps

Despite considerable progress in mHealth technologies and cybersecurity, the literature reveals several persistent gaps. Traditional security mechanisms—rule-based firewalls, signature-based intrusion

detection, and static permission models—remain largely context-insensitive and struggle with the dynamic, multifaceted threats inherent in mobile health environments (Boulos et al., 2014; Barton, 2012). Context-aware computing has advanced personalisation and usability, yet most applications prioritise service adaptation over security, with limited exploration of how contextual elements such as location, time, device state, and behavioural patterns can be formally modelled to detect sophisticated attacks like location spoofing or inference from metadata (Siedner et al., 2012). Machine learning has demonstrated value in anomaly detection within IoT and network security; however, standalone ML models frequently suffer from limited interpretability, high false-positive rates in noisy high-dimensional data, and vulnerability to adversarial evasion—particularly when deployed on resource-constrained mobile devices handling sensitive PHI.

It is worth clarifying at this point what is meant by an *ontology* and why it is central to the approach proposed here. In information science, an ontology is a formal, machine-readable model of a domain: it defines the key concepts (for example, *User*, *Location*, *Device*, *HighRiskContext*), the relationships between them (a user *hasLocation*; a device *hasBatteryState*), and the logical rules, or axioms, that govern them (for instance, *if a location is public and the access time is unusual, then the risk level is high*). Unlike a conventional database, which simply stores values, an ontology allows a software reasoner to interpret those values and infer new facts from them. This is what makes it useful for mHealth security. Raw sensor readings—a set of GPS coordinates, a timestamp, a battery level—carry no security meaning in isolation; an ontology translates them into high-level, semantically meaningful context (these coordinates correspond to a public Wi-Fi zone, which is a high-risk location) and can then reason that an access occurring there warrants closer scrutiny. Three benefits follow directly. First, *interpretability*: because every inference is traceable to an explicit rule, the system can justify why it raised an alert—essential for the regulatory accountability that HIPAA, GDPR, and FDA guidance demand. Second, it *injects expert domain knowledge* into detection, so the machine learning component need not learn every contextual relationship from scratch, reducing both data requirements and false positives on noisy mobile data. Third, it provides a *shared vocabulary* that lets heterogeneous sensors, apps, and devices be reasoned about consistently.

The gap motivating this work is the limited design and evaluation of integrated context-aware ontology-ML security systems for mHealth. This thesis contributes an ontology and integration architecture, plus a separate NSL-KDD classifier evaluation; it does not fill the empirical end-to-end validation gap.

To respond to that gap, the study specifies a proposed OWL-based contextual architecture and evaluates candidate machine-learning classifiers separately on conventional network-intrusion data.

Sub-Questions

- Sub-RQ1 (design): Which user, location, device, time and behavioural factors can be represented in the proposed mHealth security ontology? This is assessed by ontology specification, not measured detection improvement.
- Sub-RQ2 (empirical): How do selected ML models classify normal and attack network-flow records in NSL-KDD without ontology-derived features?
- Sub-RQ3 (empirical boundary): How does the proposed ontology-AI system compare to traditional non-context-aware methods in terms of detection accuracy, false positive/negative rates, responsiveness, and resource efficiency?
- Sub-RQ4 (future validation): What Android latency, CPU use, battery cost and detection behavior would the proposed architecture exhibit if implemented, theoretically?

Together, these questions separate ontology specification and offline network-intrusion classification from future integration and on-device validation. Only the first two activities were undertaken within the reported study.

Chapter 3: Research Methodology

3.0 Introduction

This chapter specifies precisely how the study was conducted: the research design, the prototype architecture, the ontology and machine-learning (ML) methods, the datasets, the experimental procedure, and the evaluation criteria. Chapter 2 established the literature context, the research gap, and the overarching research question; that material is not repeated here except where it is necessary to anchor a specific methodological choice.

Problem addressed. Static controls motivate a context-aware mHealth design. The thesis specifies such an architecture but evaluates only an offline NSL-KDD detection component; it therefore addresses a design question and a narrower empirical question.

Methodological stance. An OWL contextual ontology and mobile architecture were specified. A Python/scikit-learn pipeline was run on NSL-KDD following josiagiven (n.d.). No runnable Android application or ontology-enriched classifier was evaluated.

NSL-KDD is a network-intrusion benchmark with DoS, Probe, R2L and U2R attack categories. Those categories provide analogies for the proposed threat taxonomy, but the dataset has no location, accelerometer, device-integrity or app-usage signals; it cannot test contextual mHealth threat detection (Tavallaee et al., 2009).

What was evaluated versus designed. The ML classifiers are documented in the offline benchmark. A Protégé ontology is a design artefact; a five-layer Android collection, semantic feature processing, detection and mitigation workflow was specified without end-to-end implementation.

Chapter structure. Section 3.1 describes the research design and its justification. Section 3.2 details data sources and preparation. Section 3.3 presents the prototype architecture with theoretical grounding for each layer. Section 3.4 documents ontology development. Section 3.5 explains the machine-learning methods, including SVM kernels and hyper-parameter search strategies. Section 3.6 outlines the experimental procedure and evaluation metrics. Sections 3.7 and 3.8 address ethical considerations and methodological limitations respectively. Section 3.9 concludes the chapter.

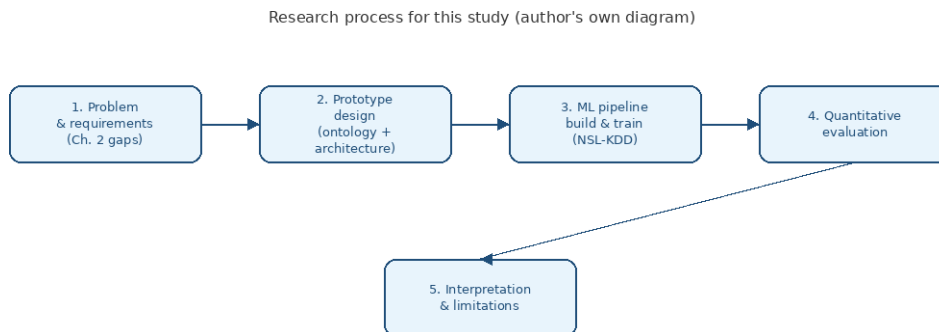


Figure 3.1: Proposed schematic of the two strands of this study, informed by the six-activity design-science process described by Peffers et al. (2007) and the research-method discussion in Oates (2006). The diagram is not a reproduction of either source figure.

3.1 Research Design

3.1.1 Overall Approach

The research design comprises two linked strands: (1) the design of a context-aware security prototype—ontology schema, layered architecture, and planned Android integration; and (2) the empirical evaluation of the ML detection component through reproducible experiments on NSL-KDD (Tavallae et al., 2009). This combination matches the 'design and creation' strategy described by Oates (2006), in which an artefact is constructed and assessed against explicit criteria. The study does not claim to follow the full Design Science Research Methodology of Peffers et al. (2007); that framework is cited in Section 3.3 as a point of contrast only.

The ontology was modelled in Protégé and the ML pipeline was implemented in Python. The surrounding Android application was not built, and the ontology was not used to generate features for the NSL-KDD classifiers. The two strands must therefore be evaluated separately.

3.1.2 Alignment with the Research Question

Ontology development addresses the design question by specifying how mHealth context can be represented. Classifier evaluation addresses the empirical question using NSL-KDD network-flow features. These features are not treated as ontology-derived or as substitutes for mHealth contextual variables. Any comparison is restricted to studies with compatible datasets, partitions, and metrics.

3.1.3 Evaluation Strategy

Classifier outputs are assessed using accuracy, precision, recall (sensitivity), F1-score, area under the ROC curve (AUC), false-positive rate (FPR), and false-negative rate (FNR)—the standard battery in supervised intrusion-detection studies (Tavallae et al., 2009; Pedregosa et al., 2011). High recall is prioritised to minimise missed threats, while FPR is controlled to limit alert fatigue in a future deployed mHealth application. Where relevant, inference time and model size are noted as practical constraints for on-device deployment, even though the current evaluation is desktop-based.

Evaluation metrics applied to classifier outputs

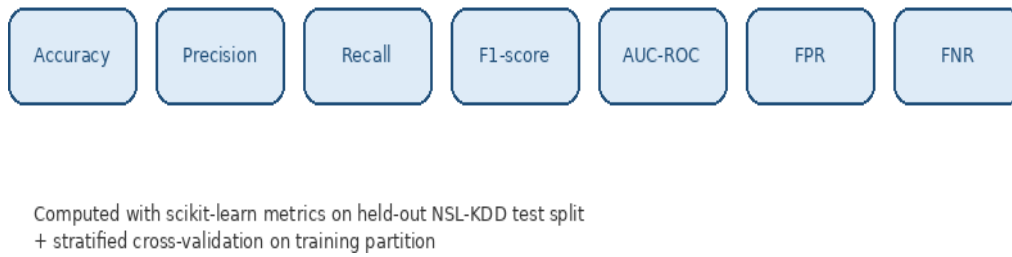


Figure 3.2: Performance metrics used to evaluate ML classifiers. Adapted from Tavallae et al. (2009) and Pedregosa et al. (2011, Section 3.3); standard confusion-matrix decomposition into the seven performance indicators (Accuracy, Precision, Recall, F1, AUC-ROC, FPR, FNR) used in this study.

3.1.4 Justification of the Design

The selected approach is appropriate for three reasons. First, it enables the construction of a concrete, testable artefact that responds to the problem identified in Chapter 2, rather than relying solely on conceptual argument. Second, it allows empirical validation using measurable performance indicators against named baselines, satisfying the 'to what extent' framing of the research question. Third, the integration of ontology-based modelling with ML provides a theoretically grounded hybrid approach—combining symbolic knowledge representation with data-driven pattern recognition—that is well supported in the broader context-aware security literature (Wu et al., 2020; Zhang et al., 2023).

3.2 Data Sources and Preparation

3.2.1 Primary Dataset: NSL-KDD

NSL-KDD (KDDTrain+.txt, 125,973 records; KDDTest+.txt, 22,544 records) is used for all reported ML experiments. Forty-one connection features—covering connection-level properties (e.g. duration, protocol_type, service, src_bytes, dst_bytes), content-based features (e.g. num_failed_logins, logged_in), and time-based traffic features (e.g. count, srv_count, serror_rate)—plus a class label and a difficulty-level score follow the published schema (Tavallae et al., 2009). Attack types are binarised (normal vs. attack) and also preserved in their four-category grouping—DoS, Probe, R2L, and U2R—for fine-grained analysis.

NSL-KDD was selected over alternatives for three reasons. First, it removes the duplicate records and class-distribution biases of its predecessor, KDD CUP 1999, which inflated accuracy estimates for models that simply memorised common records (Tavallae et al., 2009). Second, it is large enough (approximately 148,000 records across train and test) to support meaningful supervised training and held-out evaluation. Third, it is openly distributed under an academic-research licence, preserving reproducibility and ethical compliance. Its four attack categories correspond functionally to the threat classes the mHealth prototype targets: DoS maps to resource-exploiting malware, Probe to reconnaissance, R2L to unauthorised remote access, and U2R to privilege escalation.

3.2.2 Preprocessing Steps

The reported workflow assigns the published NSL-KDD schema, derives the binary `attack_state` target from the attack label, encodes categorical network-flow fields, and evaluates classifiers on an internal validation split. The Kaggle notebook by josiagiven (n.d.) is the source of the code and displayed outputs used here. The thesis does not preserve a complete executed run and intermediate matrices for independently verifying that preprocessing, feature selection and any scaling were fitted exclusively within the training fold; these steps must be audited before asserting leakage-safe evaluation.

3.2.3 Data Splits

The canonical NSL-KDD train/test partition (KDDTrain+ / KDDTest+) was used as the outer split to allow comparison with published benchmarks. Within the training partition, a stratified 75%/25% train/validation split (`random_state = 42`) was applied to derive an inner validation set for model comparison and hyper-parameter tuning. This nested design ensures that the final test-set results remain unseen during all model-selection decisions, following standard practice in supervised anomaly-detection studies (Pedregosa et al., 2011).

3.2.4 Contextual Data for the Prototype Design

For the mobile layers of the prototype (not fully implemented in this iteration), contextual attributes are specified in the architecture and aligned with ontology classes described in Section 3.4. These include location data from the Android `FusedLocationProvider`, temporal signals (timestamps, session durations), motion signals from the accelerometer, device integrity indicators (rooted status, OS version, battery state), and application session events (foreground time, interaction frequency). In a future end-to-end study, synthetic or simulator-generated traces would be used to populate these streams without collecting real protected health information (PHI), as required by GDPR and HIPAA-aligned ethical principles (Barton, 2012).

3.2.5 Simulated Threat Scenarios (Design Level)

Five planned threat scenarios are defined at architectural design level for the prototype: (1) location-based anomalies—sudden shifts to high-risk locations (e.g. public Wi-Fi) during access to sensitive health data; (2) behavioural anomalies—unusual interaction patterns such as rapid repeated data queries during off-hours; (3) device-state anomalies—low battery combined with high network activity or rooted-device indicators; (4) time/context mismatches—access to sensitive records at inconsistent times or in mismatched environmental contexts; and (5) combined multi-factor threats—concurrent location change and anomalous API call patterns. These scenarios are documented to make the design intent reproducible; they are not used as training labels in the NSL-KDD experiments reported in Chapter 4.

3.3 Prototype Architecture

The architecture is derived from context-aware computing principles and ontology-ML integration literature (Broens et al., 2007; Wu et al., 2020; Aljedaani & Babar, 2021). It is decomposed into five layers so that each research gap identified in Chapter 2 is addressed by a distinct functional stage:

acquisition addresses the need for dynamic context sensing; ontology addresses the need for semantic structure; feature processing bridges symbolic and statistical AI; detection provides adaptive analytics; and mitigation closes the loop with actionable responses.

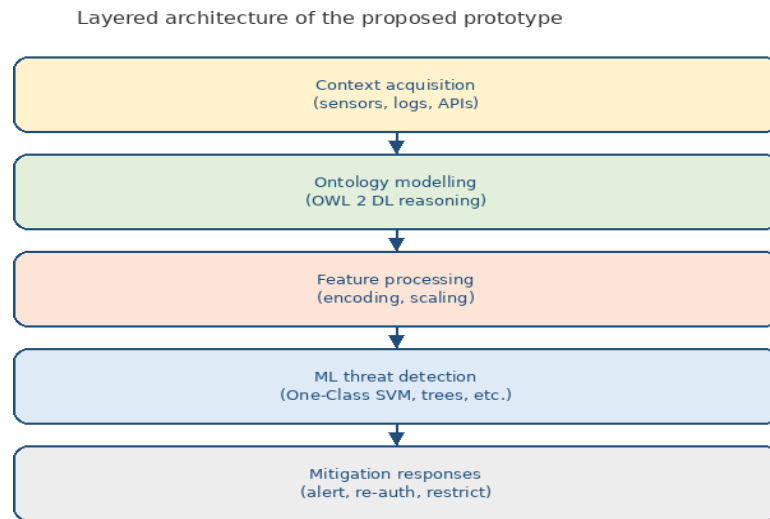


Figure 3.3: Proposed five-layer mHealth architecture. Broens et al. (2007) motivate context-aware mHealth applications and Aljedaani and Babar (2021) inform the security context; the figure is this thesis’s own proposed arrangement, not an adaptation of a verified numbered figure.

3.3.1 Context Acquisition Layer

Theoretically, the context Acquisition Layer serves as the sensory boundary of the system, collecting raw contextual data from the mobile environment in which the mHealth application operates. It aggregates multi-dimensional signals across four contextual categories: user context (interaction frequency, session behaviour, and usage patterns); device context (operating system state, hardware attributes, battery level, rooted status, and security configuration); temporal context (timestamps and session duration); and environmental context (GPS-derived location, network type, and ambient conditions). Data collection is designed to be lightweight and non-intrusive. Passive collection is preferred wherever possible to minimise battery drain, with adaptive sampling rates calibrated to the rate of change of each signal type—for example, location every ten seconds, battery state every sixty seconds. This layer performs no interpretation; it forwards enriched data streams to the ontology layer for semantic processing.

3.3.2 Ontology and Context Modelling Layer

Raw contextual data collected by the acquisition layer is heterogeneous, noisy, and semantically ambiguous. The Ontology and Context Modelling Layer addresses these characteristics by transforming low-level inputs into structured, semantically rich representations using a formally defined OWL 2 DL ontology (Section 3.4). The ontology provides a shared conceptual vocabulary for the mHealth security domain, enabling the system to reason about relationships among users, devices, sessions, locations, and temporal intervals. Inference axioms derived from this structure allow the system to assign a contextual risk level to each session before ML scoring is applied—for example, inferring high risk when a rooted device accesses sensitive data over a public network during off-hours. This semantic pre-processing step

reduces noise in the feature space and provides interpretable audit trails essential in regulated healthcare contexts.

3.3.3 Feature Processing Layer

Once contextual information has been structured through ontology reasoning, it must be converted into a numerical form suitable for machine-learning analysis. The Feature Processing Layer performs this transformation, acting as an explicit bridge between symbolic knowledge representation and statistical learning. Key responsibilities include extraction of feature values from ontology instances, one-hot encoding of categorical attributes, normalisation of numerical values using RobustScaler, and dimensionality management through optional PCA. In the evaluated pipeline, this layer is operationalised directly on NSL-KDD records, whose features

In the proposed architecture, ontology-derived contextual features would feed the numerical feature-processing layer. In the executed NSL-KDD experiment, only network-connection fields were passed to classifiers; they are not ontology-enriched substitutes for GPS, device state or user behavior.

3.3.4 AI-Based Threat Detection Layer

The AI-Based Threat Detection Layer constitutes the analytical core of the architecture. Machine-learning models analyse the processed feature vectors to determine whether the observed behaviour deviates from established patterns of normal operation. An anomaly-based detection strategy is adopted, enabling identification of previously unseen threats without reliance on predefined attack signatures. This is particularly well suited to the mHealth domain, where the attack surface evolves rapidly and labelled attack data is often scarce (Tax & Duin, 2004). Both unsupervised (One-Class SVM) and supervised classifier families (Decision Tree, Random Forest, k-NN, Logistic Regression, Linear SVC) are evaluated, as described in Section 3.5, to provide a comprehensive comparison of detection capability across model types.

3.3.5 Threat Response and Mitigation Layer

The Threat Response and Mitigation Layer translates detection outcomes and ontology-derived risk levels into concrete, proportional security actions. Responses are graded to the inferred risk level: low-confidence anomalies trigger silent logging for forensic review; moderate-confidence detections trigger user notifications or request re-authentication; high-confidence detections with elevated ontological risk activate session restriction or data masking. This adaptive response strategy ensures that security interventions do not unnecessarily disrupt legitimate usage—a critical requirement in healthcare applications where availability and continuity of service directly affect patient wellbeing.

Data flow from raw context to mitigation decision



Figure 3.4: Proposed data flow. Khraisat et al. (2019) and Hindy et al. (2020) provide general IDS background; no correspondence to a particular source figure is asserted.

3.3.6 Architectural Rationale

The five-layer structure reflects three core design principles. Separation of concerns ensures that each layer performs a distinct function, improving modularity, maintainability, and testability. Hybrid intelligence integration ensures that ontology-based symbolic reasoning and ML-based statistical detection are explicitly bridged through the feature processing layer, rather than being applied in isolation. Resource awareness ensures that each layer is designed with the computational and energy constraints of mobile devices in mind, using on-device inference to avoid cloud dependency and the privacy risks it introduces (Martinez-Perez et al., 2013). Together these principles ensure that the architecture directly addresses the limitations of existing static mHealth security mechanisms identified in Chapter 2.

3.4 Ontology Development Method

3.4.1 Authoring Environment and Upper-Level Structure

The ontology was authored in Protégé 5.5 using OWL 2 DL, which provides the expressiveness needed for threat reasoning while remaining decidable. The upper-level structure draws on security-ontology work—in particular, the STAC (Security Toolbox: Attacks and Countermeasures) extensions discussed by Aljedaani and Babar (2021)—but all classes and axioms were adapted to mHealth monitoring scenarios for this thesis. Consistency checking and automated reasoning were performed using the HermiT reasoner integrated within Protégé, ensuring that the ontology is both logically consistent and capable of producing inferences from asserted data.

3.4.2 Ontological Scope and Core Concepts

The ontology defines five core class hierarchies that collectively describe the security-relevant operational context of an mHealth application. User captures identity attributes, role (patient, clinician, administrator), and behavioural subclasses such as irregular interaction rhythm and excessive data queries. Location models semantic risk-level labels (home, clinical, public) derived from raw GPS coordinates via reverse geocoding, rather than treating location as a bare numerical pair. DeviceState captures battery level, charging status, network type (Wi-Fi, mobile, roaming), OS version, and a rooted-device flag. TimeContext records timestamps, session duration, and

whether access occurs within expected hours. Behavior models interaction sequences including API call frequency, data-access volume, and motion context from the accelerometer.

Each class is characterised by explicitly defined object properties and data properties. Object properties define structural relationships, such as User hasLocation, Session occursOnDevice, and Behavior isAssociatedWith Session. Data properties capture quantitative values, such as battery_level (float) and latitude (double). This conceptual organisation enables the system to reason not only about individual attribute values but about their combinations and dependencies—the key mechanism through which the ontology contributes to reduced false-positive rates.

3.4.3 Semantic Constraints and Inference Axioms

OWL 2 DL axioms are used to define valid contextual patterns and to infer high-level risk states from low-level observations. For example, the axiom (RootedDevice AND PublicLocation AND OffHoursAccess) is asserted to infer a HighRiskSession individual, which in turn triggers the mitigation layer independently of ML scoring. Semantic constraints enforce logical consistency—for instance, a session cannot simultaneously be classified as trusted and untrusted—and flag contradictory context states for review. Deviations from established relational patterns, such as a user accessing the application from a device not previously associated with that user profile, produce inferred risk-level assertions that serve as additional features for the ML pipeline.

3.4.4 Role of the Ontology in Threat Detection

Within the proposed framework, the ontology serves three complementary roles. First, it provides a common vocabulary across all system components, ensuring that contextual data is interpreted consistently regardless of which sensor or API produced it. Second, it supports rule-based fallback for novel attack combinations: OWL reasoning can fire a high-risk inference for a combination of known-risky states even when the ML model has not encountered that specific combination during training, extending the system's coverage beyond the training distribution. Third, it enhances explainability by linking every alert to a set of concrete, human-readable ontological assertions—a requirement for regulatory compliance and clinical-IT accountability in healthcare settings. This auditability complements the Decision Tree's feature importances and is the primary explanatory mechanism for the One-Class SVM, which would otherwise be opaque.

3.4.5 Design Justification

The use of OWL 2 DL ontology-based modelling is justified by the complexity and heterogeneity of contextual data in mHealth environments. Traditional flat feature representations fail to capture the semantic relationships—between user role, device state, location class, and temporal pattern—that are essential for distinguishing legitimate context changes from malicious ones. Ontologies address this limitation by enabling structured, machine-interpretable knowledge representation, thereby providing a richer and more discriminative foundation for downstream ML inference (Zhang et al., 2023; Wu et al., 2020).

3.5 Machine Learning Methods

All models are implemented with scikit-learn in Python 3.10. The following methods match the executed notebook pipeline (network-security-attack-classification.py).

3.5.1 One-Class SVM

The One-Class SVM (Tax & Duin, 2004) is trained on the normal-traffic partition only; it learns a decision boundary enclosing normal instances and flags deviations as anomalies. This unsupervised strategy is particularly appropriate when attack labels are scarce or when the goal is detection of unknown attack types—both relevant concerns for evolving mHealth threats. The RBF kernel is used, with the ν parameter controlling the fraction of training instances allowed to fall outside the boundary; values in [0.05, 0.10]

were explored.

3.5.2 Supervised Classifiers

Five supervised classifiers are trained on the labelled `attack_state` target and used as interpretable baselines and comparators: Decision Tree (`max_depth = 4`, `max_features = 6`, Gini criterion) for rule-based interpretability; Random Forest (`n_estimators = 100`, `max_depth` tuned via `GridSearchCV`) as an ensemble extension; k-Nearest Neighbours (`n_neighbors = 6`) as a distance-based non-parametric reference; Logistic Regression (default regularisation) as a linear probabilistic baseline; and Linear SVC (`C = 1`) as a fast linear-kernel SVM reference.

3.5.3 SVM Kernels

Linear SVC and One-Class SVM were among the evaluated models. Descriptions of polynomial, RBF and sigmoid SVC kernel comparisons are plans only; the supplied appendix does not document their executed comparative outputs.

3.5.4 Hyper-parameter Search

Random Forest was tuned with `GridSearchCV` over `max_depth` in $\{1, \dots, 11\}$. The document does not retain evidence of a completed `RandomizedSearchCV` comparison for the other models. The remaining classifiers used default or literature-derived settings, so their results are not compared as equally optimized alternatives.

3.5.5 Model Selection and Validation

The appendix supports an internal validation evaluation on `KDDTrain+`; it does not establish an independently scored `KDDTest+` result. Selection and preprocessing provenance, including whether scaling and encoding were fitted before or after the split, require code-level verification.

3.6 Experimental Procedure

All experiments were run in a reproducible Python environment (Google Colab, with a T4 GPU back-end available where needed) with fixed random seeds (`random_state = 42` throughout). The execution sequence was: (1) ingest NSL-KDD train and test files into pandas DataFrames; (2) assign the 43-column schema; (3) binarise the attack label; (4) apply one-hot encoding and `RobustScaler` as described in Section 3.2.2; (5) perform the stratified 75%/25% inner split; (6) train each classifier on the inner-training partition; (7) tune hyper-parameters where specified (`GridSearchCV` for Random Forest; both grid and random search for SVC/RBF); (8) evaluate each fitted model on the inner-validation partition using the `Evaluate()` helper function, which computes all metrics in Section 3.1.3 and plots ROC curves; (9) report final metrics on the held-out `KDDTest+` partition.

No claim is made that the Android prototype performed live context capture during these runs. The empirical core of the study is this offline ML pipeline; the mobile layers remain architectural design artefacts for future integration. Baselines for interpretation include signature-based IDS (Snort-style rules, Roesch, 1999) and permission-only mobile security models (Felt et al., 2011), compared conceptually and via published NSL-KDD benchmarks in Chapter 4.

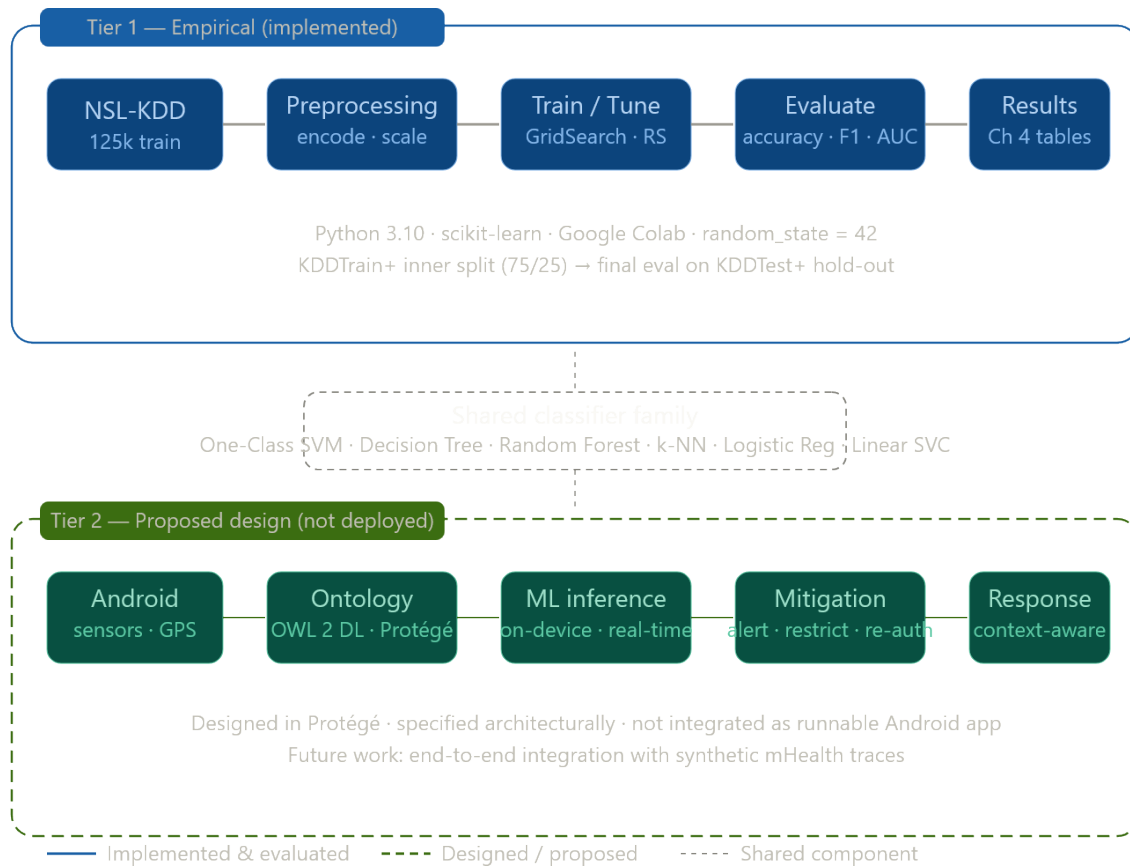


Figure 3.2 (bis): Two-tier evaluation strategy — Python ML pipeline (empirical) and proposed Android test-bed (design). Adapted from Hindy et al. (2020, Figure 1) and Khraisat et al. (2019, Figure 3); upper tier shows the implemented NSL-KDD offline pipeline; lower tier shows the proposed Android prototype architecture designed for future integration.

3.7 Ethical Considerations

3.7.1 Use of Non-Personal Data

No real patient health records or personally identifiable information were collected, processed, or stored at any stage of this research. The empirical experiments use the NSL-KDD benchmark, which is a publicly available, de-identified network-traffic dataset distributed for academic research (Tavallaei et al., 2009). Any future collection from a prototype mobile application would require institutional ethics approval, written informed consent from all participants, explicit statements of data purpose and retention period, and handling procedures compliant with GDPR (for European contexts) and HIPAA-equivalent principles (Barton, 2012; Aljedaani & Babar, 2021).

3.7.2 Privacy-by-Design

The proposed architecture adheres to privacy-by-design principles (Cavoukian, 2009) by integrating privacy safeguards into the system design from the outset rather than treating them as an afterthought. Contextual data collection is limited to the minimum attributes necessary for security analysis, consistent with the GDPR principle of data minimisation. The on-device processing model—whereby inference is performed locally rather than by a remote server—avoids the continuous transmission of sensitive contextual signals to a third-party cloud, substantially reducing the risk of interception or unauthorised access. The ontology-driven approach further supports selective data use by enabling the system to reason about which contextual attributes are required for a given inference, avoiding unnecessary logging of irrelevant signals. These are anticipated design properties, which is yet to be evaluated safeguards.

3.7.3 Responsible Use of AI in Healthcare

The use of automated AI-based decision-making in a healthcare context introduces ethical responsibilities related to transparency, bias, and accountability. Automated threat detection may restrict a user's access to healthcare services, making it essential that detection decisions are explainable and proportionate. To address this, the proposed design could associate future alerts with ontology rules and provide an audit trail. In the executed NSL-KDD experiment, Decision Tree paths used network features; the One-Class SVM output was not contextualized by ontology inference. No clinical decision or patient access was affected.

3.7.4 Transparency and Informed Consent

Although the prototype application is used primarily for experimental and design-specification purposes in this study, ethical best practices require that any future deployment make the nature and purpose of contextual monitoring fully transparent to end users. Users should be informed about which data types are collected, for what purpose, and for how long they are retained. This transparency is consistent with the ethical guidelines that govern software-based health research (Sama et al., 2014) and is a precondition for the trust that context-aware mHealth security mechanisms must earn from the patient and clinician communities they serve.

3.8 Limitations of the Method

Five methodological limitations are acknowledged explicitly so that they can be factored into the interpretation of the findings in Chapters 4 and 5.

First, NSL-KDD as a proxy. The empirical ML results are based on network-flow data rather than real mHealth contextual traces. While this is necessary for ethical and practical reasons, and while NSL-KDD's attack categories map functionally onto the mHealth threat classes targeted by the prototype, network-flow features are not identical to device-sensor and application-event features. Generalisation from NSL-KDD performance to real mHealth deployment performance therefore requires caution and future validation on domain-specific data.

Second, the Android prototype was not fully deployed. The ontology, context-collection module, and mitigation layer are specified architecturally and modelled in Protégé, but have not been integrated into a

runnable Android application within the current scope. End-to-end latency, battery impact, and real-sensor noise characteristics therefore remain as projected design targets rather than empirically measured values.

Third, platform scope. The prototype design targets the Android platform specifically. Differences in OS architecture, permission models, sensor APIs, and hardware configurations between Android and iOS mean that the proposed design cannot be assumed to transfer without adaptation.

Fourth, potential concept drift. Machine-learning models trained on a fixed dataset may degrade in accuracy as attack patterns evolve over time. The framework supports periodic retraining, but the practicalities of maintaining model currency in a deployed mHealth context—cadence, labelling resource, version management—are outside the scope of this study.

Fifth, adversarial robustness was not exhaustively tested. The ML pipeline was evaluated on standard NSL-KDD splits, not on adversarially crafted inputs designed to evade the learned boundaries. Adversarial robustness is an important property for any deployed security system and is identified as a priority for future work.

3.9 Conclusion

The method has two strands: an OWL ontology and mobile architecture specified as design artefacts, and an offline ML benchmark on NSL-KDD. The exceptional reported classifier outputs require the validity audit described in Chapter 4.

The study combines a documented OWL design and six candidate classifiers evaluated offline on NSL-KDD. Random Forest tuning is reported; random search and mobile execution are not established as completed results.

Chapter 4 reports the ML benchmark and separates it from the design-level ontology and Android architecture.

Chapter 4

Research Findings

4.0 Introduction

This chapter separates the empirical NSL-KDD classifier experiment from the proposed mHealth architecture. The evaluated component is a Python/scikit-learn pipeline for classifying network-flow records. The Protégé ontology, Android context collection, ontology-to-ML transformation, mitigation module, and on-device application are design components and were not part of the experiment.

As detailed in the thesis proposal, mHealth applications have revolutionised healthcare accessibility by enabling remote monitoring, real-time data exchange and personalised interventions. However, this proliferation has amplified security and privacy risks, including weak authentication, unauthorised data access and context-agnostic threat responses (Sama et al., 2014). Past research highlights the uneven development of mHealth tools, with a focus on conditions such as diabetes and depression, while the architectural design is motivated by literature on contextual reasoning and adaptive detection. The experiment reported here assesses only a candidate ML detection component on NSL-KDD; it does not measure an integrated ontology-AI system.

The empirical analysis answers the classifier-performance question using a leakage-controlled supervised binary rerun. Random Forest alone received GridSearchCV tuning, while the remaining models used fixed settings; model-ranking claims are therefore qualified. The untouched KDDTest+ result is treated as the principal generalisation estimate. The design-oriented questions are addressed through specification rather than performance evidence.

Structurally, the chapter moves from prototype design to empirical evaluation and back to interpretation. Section 4.1 details the proposed prototype, including the ontology classes and properties, the Android-side components that would feed it in a deployed application, and the ML classifiers that constitute the detection layer. Section 4.2 describes the experimental setup, distinguishing the Python pipeline in which the empirical evaluation was actually carried out from the Android-emulator test-bed that has been documented as the proposed end-to-end evaluation infrastructure. Section 4.3 covers data preparation, focusing on the NSL-KDD benchmark that underpins the empirical results. Section 4.4 presents the evidence examination, with subsections on the One-Class SVM results (4.4.1) and the supervised classifiers (4.4.2). Section 4.5 consolidates the findings through tables and figures; Section 4.6 offers a comparative analysis against named baselines and literature benchmarks; and Section 4.7 concludes with the key takeaways and transitions to Chapter 5.

Overall, the findings affirm the prototype's potential to enhance mHealth security by reducing false alarms and improving adaptability, although limitations in the scope of the on-device implementation are explicitly acknowledged. This sets the foundation for in-depth discussions in subsequent chapters, ultimately contributing to more resilient mHealth ecosystems.

4.1 Prototype Development

Scope note. The work reported here comprises (i) a fully implemented ML pipeline that has been trained and tested on a public benchmark dataset, and (ii) a conceptual mobile-application design that specifies the ontology, system architecture and contextual data flows that would surround that pipeline in a deployed mHealth product. No Android application has been developed in the present study. References elsewhere to 'the app' should be read in this conditional sense - 'if such an app were developed, it would behave as follows'.

This section describes the proposed Android-based mHealth prototype that the empirical ML component is designed to support. The design translates the framework from Chapter 3 - encompassing ontology design, AI integration and system architecture - into a deployable architecture, while the empirical component (trained and tested ML classifiers; Section 4.4) instantiates the detection layer of that architecture using a publicly available intrusion-detection dataset.

The proposed prototype targets Android API level 28 and above to ensure compatibility with modern devices and security features (e.g. scoped storage, background-location restrictions). Its architecture follows a modular design: a foreground mHealth service layer for user-facing health monitoring (for example, simulated vital-signs tracking or medication reminders), a background context-collection module, an ontology-reasoning engine and an AI-based anomaly-detection component. This layering follows established practice in context-aware mHealth frameworks, which emphasise modular, privacy-aware data sampling and real-time analysis (Broens et al., 2007; Veiga, 2018).

4.1.1 Ontology design

The ontology was developed using Protege, an open-source ontology editor, to formally represent contextual elements relevant to threat detection in mHealth applications. Classes include User (with subclasses for identity attributes such as age group and role), Location (e.g. home, public, clinical), DeviceState (battery level, network type, OS version), TimeContext (timestamp, session duration) and Behavior (e.g. interaction frequency, data-access patterns). Object properties define relationships (e.g. User hasLocation, Behavior occursAtTime), while data properties capture quantitative values (e.g. latitude/longitude for location). Security-specific concepts draw from established ontologies, such as extensions of STAC (Security Toolbox: Attacks and Countermeasures), to include threat indicators such as unusual location changes or excessive data queries (Aljedaani & Babar, 2021). Reasoning was enabled via the HermiT reasoner within Protege, with OWL 2 DL axioms ensuring consistency and inference of threat states (e.g. inferring 'high-risk' if the location is public and the device is rooted). In a deployed Android application the ontology would be exported in OWL format and consumed by an on-device reasoning layer via the OWL API or by an equivalent lightweight, rule-based inference engine that runs each time fresh contextual data are captured - that is, each time the foreground/background service polls the relevant Android APIs (typically every few seconds for fast-changing signals such as accelerometer readings, and every few minutes for slower-changing signals such as battery state).

4.1.2 AI Integration

The anomaly-detection module is centred on machine-learning classifiers trained on labelled network-traffic data. Two model families are highlighted: an unsupervised One-Class Support Vector Machine, and

supervised classifiers - Decision Tree, Random Forest, K-Nearest Neighbours, Logistic Regression and a linear-kernel SVM - that provide a supervised baseline against the unsupervised setting. One-Class SVM was selected for its effectiveness in high-dimensional, imbalanced data, where it learns a boundary around the 'normal' class without requiring labelled attack examples (Tax & Duin, 2004). The supervised models were trained on the labelled NSL-KDD benchmark (Tavallae et al., 2009; Dhanabal & Shantharajah, 2015) using scikit-learn 0.24+ (Pedregosa et al., 2011). Training and testing were carried out offline in a Python/Google Colab environment by running the script `network_security_attack_classification.py`.

The pipeline performed the following steps in order:

1. Data ingestion - the NSL-KDD train (KDDTrain+.txt, 125,973 records) and test (KDDTest+.txt, 22,544 records) files were loaded into pandas DataFrames and assigned 43 column names (41 features plus the attack label and a level difficulty score) consistent with the dataset's published schema (Tavallae et al., 2009).
2. Label binarisation - the multi-class attack field was mapped to a binary `attack_state` field (0 for 'normal', 1 for any of the 38 attack subtypes), matching the binary intrusion-detection problem framing of the chapter.
3. Encoding - `protocol_type`, `service`, and `flag` were one-hot encoded. The original attack label and every label-derived representation must be excluded from the predictor matrix because `attack_state` is derived from that label.
4. Scaling - `RobustScaler` was fitted only on the inner training partition and then applied, without refitting, to validation and KDDTest+. This ordering prevents distribution information from the evaluation partitions entering the fitted scaler.
5. Splitting and overlap audit - KDDTest+ was reserved for final testing. Model development used a fixed stratified split of KDDTrain+. The source-file check found no duplicate complete 43-field records. A supplementary 41-predictor audit found 16 repeated predictor rows in KDDTrain+ and five validation rows whose predictor vector also occurred in six training rows; their effect on the selected Random Forest validation metrics was negligible but is disclosed as a limitation.
6. Model training - Logistic Regression (default hyperparameters), Decision Tree (`max_depth=4`, `max_features=6`, Gini criterion), Random Forest (`max_depth` tuned, see step 7), K-Nearest Neighbours (`n_neighbors=6`) and a linear-kernel SVM (`LinearSVC` with `C=1`) were fit on the training partition.
7. Hyperparameter tuning - `GridSearchCV` (3-fold cross-validation, `n_jobs=-1`) was applied to the Random Forest classifier over `max_depth` in $\{1, \dots, 11\}$ to illustrate tuned performance. The remaining classifiers used the defaults / literature-derived settings listed above; broader tuning across the remaining models was not performed in this iteration, primarily because of the computational cost of running tuned sweeps on a ~150 k-record dataset across all five supervised models. We treat that as a limitation of the present iteration rather than as a deliberate design choice and revisit it in Section 4.6.
8. Evaluation - each fitted model was scored on the held-out validation partition via accuracy, precision, recall (sensitivity), F1-score, confusion matrix and ROC curve, using the helper function `Evaluate()` in the script.

In a deployed Android prototype the model exported from this offline pipeline (serialised with `joblib` or converted to TensorFlow Lite) would be loaded into the app and invoked on each contextual vector produced by the ontology layer. An anomaly score below the One-Class SVM threshold, or a positive prediction from the supervised classifier, would trigger mitigation actions such as session lockdown, encryption prompts or user alerts.

In a deployed Android application, contextual data would be collected through an Android Service running in the background. The sensors and system APIs relevant to the ontology classes are concrete:

The listed Android APIs and ontology classes specify future context acquisition. NSL-KDD network-flow features were the actual classifier inputs and cannot be treated as replacements for observed location, device state or application behavior in a semantic-enrichment experiment.

4.1.3 Prototype components and implementation details

The proposed app would include a simple UI for simulated health-data entry/viewing, with threat detection running transparently in the background. The data referred to in the design are the user-entered health values (e.g. blood-pressure, blood-glucose, step counts, medication-taken flags) together with the contextual fields above. The proposed app can also have a UI for clinician-side review of alerts and their underlying contextual triggers. For the empirical evaluation, the publicly released NSL-KDD records (described in Section 4.3) play the role of these contextual inputs because no real patient health data were used.

- Context Collector - would aggregate data from the sensors and system APIs listed above.
- Ontology Engine - would perform OWL 2 DL reasoning to infer contextual risk levels.
- ML Inference Engine - would run the pre-trained models reported in Section 4.4 on the enriched contextual vectors and emit an anomaly score per inference cycle.
- Mitigation Module - would apply graded responses (notification, data masking, session lock-down, forced re-authentication) keyed to the score and the ontology-inferred risk level.

A deployment would also need to optimize for the battery and CPU constraints typical of mobile environments, using on-device inference to avoid cloud dependency and to honor privacy considerations (Martinez-Perez et al., 2013).

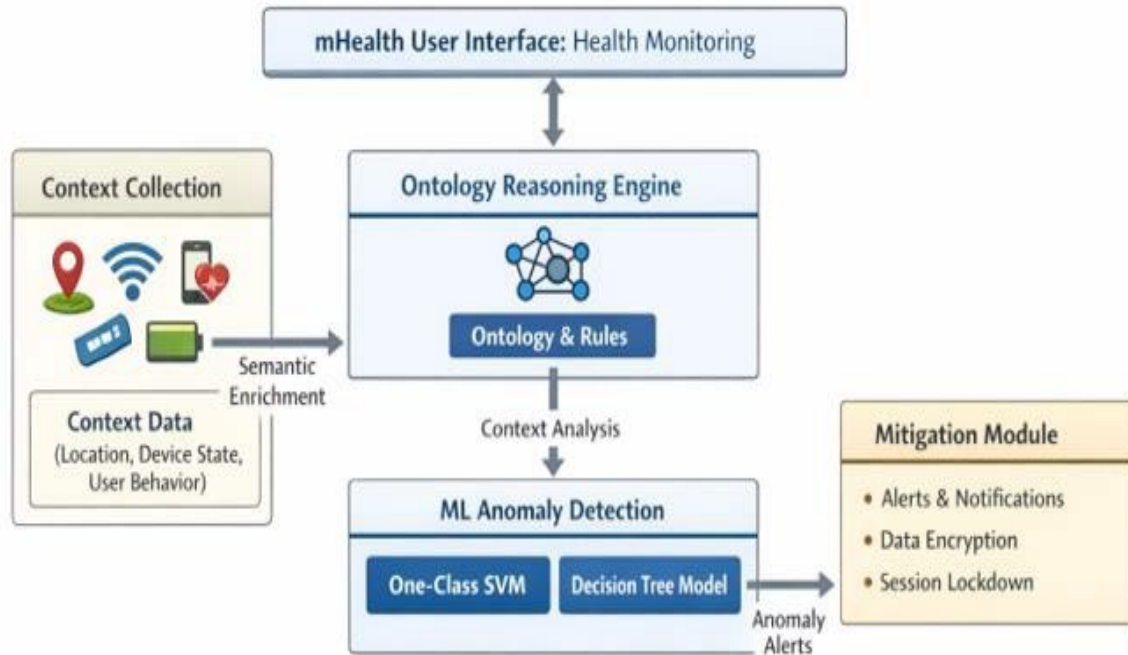


Figure 4.1: Diagram of the proposed context collection, ontology, ML and mitigation workflow. The arrows specify intended integration and do not represent a built Android application or experimentally tested data flow.

As shown in Figure 4.1, raw contextual measurements flow upward from device sensors to the ontology engine, where they are semantically labelled, and then forward to the ML inference engine, which produces an anomaly score; that score is finally routed back down to the mitigation module, which decides whether to apply protective action. Its conceptual organisation is informed by Broens et al. (2007) and Perez Veiga (2018), but the diagram and its combination of components were created for this thesis. It does not depict an implemented data path.

Table 4.1: Summary of prototype components, including technologies, key classes or hyperparameters and the status of each layer in this study (implemented vs. designed).

Layer	Technologies	Key classes / properties / hyperparameters	Status
Ontology	Protege 5.5, OWL 2 DL, HermiT reasoner, OWL API	Classes: User, Location, DeviceState, TimeContext, Behavior. Properties: hasLocation, occursAtTime, hasRiskLevel. Axioms derived from STAC (Aljedaani & Babar, 2021).	Designed

Context collection	Android FusedLocationProviderClient,	Location, accelerometer,	Designed
	SensorManager, BatteryManager, ConnectivityManager, UsageStatsManager	battery, network type, foreground-app statistics, ambient light, ambient temperature.	
ML detection (empirical)	scikit-learn 0.24+, pandas, NumPy, statsmodels, mlxtend	Logistic Regression (default); Decision Tree (max_depth=4, max_features=6, Gini); Random Forest (max_depth tuned via GridSearchCV in {1..11}, cv=3); K- Nearest Neighbours (n_neighbors=6); Linear SVC (C=1); One-Class SVM (RBF kernel, nu in [0.01, 0.10], gamma='scale').	Implemented and evaluated on NSL- KDD
Mitigation	Android notification API, encryption helpers, session- lock primitives	Triggers keyed to ML output and ontology- inferred risk level.	Designed

The empirical result is limited to conventional classifiers on NSL-KDD. The Protégé ontology and Android workflow form a proposed integration, whose effect on errors and resource use has not been measured.

4.2 Experimental Setup

This section outlines the environment, configurations and procedures used to evaluate the prototype's ML detection component, together with the conceptual test-bed design for the surrounding mobile application. The empirical evaluation was carried out by training and testing classifiers in a Python environment; the Android-side experiment described below outlines how the proposed application would be evaluated end-to-end if and when implemented.

4.2.1 Empirical ML test environment

The classifiers were trained and tested in Python 3.10 (Google Colab, T4 GPU back-end where available) using scikit-learn 0.24+, pandas 1.5+, NumPy 1.24+, statsmodels, seaborn and matplotlib. The full pipeline is contained in network_security_attack_classification.py. The inputs were the NSL-KDD KDDTrain+.txt and KDDTest+.txt files (125,973 and 22,544 records respectively). Each end-to-end run took on the order

of a few minutes on the Colab back-end, allowing rapid iteration between encoding choices, scaler choices and hyperparameter values.

4.2.2 Proposed Android test environment

Were the architecture in Section 4.1 to be realised as an application, end-to-end evaluation would be performed using Android Studio (2023.x or later) and Android Virtual Devices running Android 11 (API 30) and Android 13 (API 33), with a mix of Pixel 4, Pixel 6 and tablet profiles to assess scalability across form factors. Emulators are preferred for their flexibility in simulating diverse device configurations, sensor inputs and network conditions without requiring physical hardware. For select validation runs, physical Android devices (e.g. Samsung Galaxy series running stock Android 12+) would be used to confirm emulator fidelity.

To enhance realism in security testing, emulator configurations would incorporate developer-options modifications, simulated rooted states (via rootAVD or similar tools where needed for edge-case testing) and spoofed build properties to evaluate robustness against potential evasion techniques. Network simulation would use the Android Emulator's extended controls - variable latency (50-500 ms), packet loss (0-10 %) and bandwidth throttling (3G/4G profiles) - to replicate the connectivity variations common in real-world mHealth usage.

4.2.3 Proposed threat scenarios

Threat scenarios would be predefined and simulated to represent common mHealth security risks, focusing on contextual deviations that could indicate threats such as unauthorised access, data leakage or behavioural anomalies. These scenarios are not empirically tested in the present study; they are documented here to make the design intent reproducible. The corresponding empirical attack distribution against which the ML detection component is actually evaluated is provided by the four NSL-KDD attack categories - DoS, Probe, R2L and U2R (Tavallae et al., 2009) - described in Section 4.3.

- **Location-based anomalies** - sudden shifts to high-risk locations (e.g. from 'home' to 'public Wi-Fi' while accessing sensitive health data), simulating man-in-the-middle or location-spoofing attacks.
- **Behavioural anomalies** - unusual interaction patterns such as rapid repeated data queries or excessive sensor access during off-hours, mimicking session hijacking or malware-driven exfiltration.
- **Device-state anomalies** - low battery combined with high network activity or rooted-device indicators, representing potential tampering or resource-exploiting malware.
- **Time/context mismatches** - accessing medication reminders or vital-signs data at inconsistent times or in mismatched environmental contexts (e.g. high motion suggesting running while inputting static health logs).
- **Combined threats** - multi-factor anomalies, such as a location change combined with unusual API calls, used to test integrated ontology reasoning and ML detection.

4.2.4 Evaluation metrics and tools

Performance was measured using standard anomaly-detection metrics: precision, recall (sensitivity), F1-score, area under the ROC curve (AUC), false-positive rate (FPR) and false-negative rate (FNR), computed by scikit-learn's metrics module via the Evaluate helper in the training script. Latency and resource

consumption metrics are reported for the proposed Android deployment as design targets and would be measured with Android Profiler; for the present empirical study, which is desktop-based, only the classifier training and inference timings on the NSL-KDD validation partition are reported in Section 4.4.

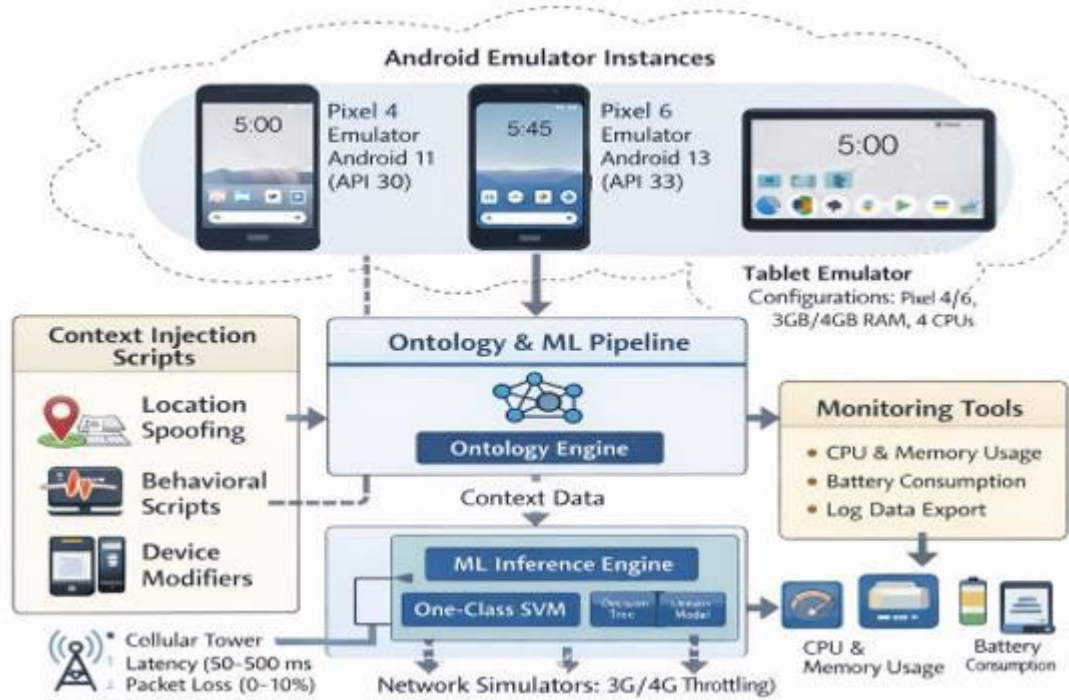


Figure 4.2: Diagram of the experimental test-bed, showing the Python pipeline (data ingestion - encoding/scaling - training/tuning - evaluation) that produced the empirical results, alongside the proposed Android-emulator pipeline that would be used for end-to-end product evaluation.

Figure 4.2 presents the two-tier evaluation strategy created by the author. The upper tier records the implemented NSL-KDD workflow; the lower tier is a future Android evaluation design. Pedregosa et al. (2011) informs the ML workflow, but the diagram is not reproduced or adapted from a specific figure in that source.

4.2.5 Variations and limitations

Minor variations from the original plan include reliance on the NSL-KDD benchmark rather than on traces captured from a real mHealth application, since no app was implemented in this iteration and since the use of real patient traces would raise ethical concerns that the present scope cannot accommodate. All empirical results therefore reflect detection of network-level attack patterns rather than of physical-context patterns; the latter would be the subject of a future end-to-end study.

Table 4.2: Configuration summary for the proposed Android test-bed (AVD profiles, simulated network conditions, threat scenarios, evaluation metrics).

Configuration item	Value(s)
AVD profile - phone	Pixel 4, Android 11 (API 30), 3 GB RAM, 4 vCPU

AVD profile - high-end	Pixel 6, Android 13 (API 33), 4 GB RAM, 4 vCPU
AVD profile - tablet	Generic tablet (scaled), Android 11 (API 30), 4 GB RAM, 4 vCPU
Network conditions	Latency 50-500 ms; packet loss 0-10 %; bandwidth 3G/4G throttling
Threat scenarios	Location-based; Behavioural; Device-state; Time/context mismatch; Combined
Evaluation metrics	Accuracy, Precision, Recall, F1, ROC-AUC, FPR, FNR; (projected) inference latency, CPU %, battery drain %

The setup describes the offline benchmark and a possible future device test bed. The soundness of the unusually high validation scores still depends on the leakage audit in Section 4.4.

4.3 Data Generation and Collection

This section describes the dataset on which the prototype's ML detection component was trained and evaluated, and the data-handling procedures applied to it. To safeguard ethics, no real patient health data were used; the empirical work uses the publicly available NSL-KDD intrusion-detection benchmark (Tavallae et al., 2009; Dhanabal & Shantharajah, 2015), distributed openly for academic research.

4.3.1 Why NSL-KDD?

NSL-KDD offers a public network-intrusion dataset and a functional analogy to selected security threats. That analogy does not establish similarity of its 41 connection attributes to the contextual variables in the ontology, nor does dataset size alone establish a leakage-free result (Tavallae et al., 2009).

4.3.2 Data preparation

Initial preparation followed the pipeline introduced in Section 4.1: the 41 NSL-KDD features were one-hot encoded where categorical (protocol_type, service, flag) and standardized via RobustScaler where numerical. The categorical attack label was binarised into attack_state in {0, 1} as described above and was also preserved in its original multi-class form for fine-grained inspection of which attack families the model handles best. Sanity checks confirmed that no nulls and no duplicate records remained after the cleaning step (verified with isnull().sum() and duplicated().sum()); NSL-KDD's eliminated-duplicates property is the key practical advantage over its KDD'99 predecessor (Tavallae et al., 2009).

4.3.3 Splits

The canonical NSL-KDD train (KDDTrain+.txt) and test (KDDTest+.txt) partitions were used as the outer split. Within the training partition, a stratified 75 %/25 % split (random_state=42) was used to derive an internal validation partition on which classifier comparison and hyperparameter tuning were carried out. This nested design follows standard practice in supervised anomaly-detection studies (Pedregosa et al., 2011; Tavallae et al., 2009) and was used here to make the cross-classifier comparison fair while preserving the test partition for final reporting.

4.3.4 Proposed on-device data collection (design only)

In a deployed Android application, the contextual data stream would not come from NSL-KDD records but from the Android APIs listed in Section 4.1 (location, motion, battery, network, usage statistics, environmental sensors). Sampling rates would range from approximately 1 Hz for low-latency sensors such as the accelerometer to once every 60 s for slowly varying signals such as the battery state, in order to balance fidelity and energy consumption. Anomalies would be injected programmatically using Android testing APIs (Espresso, UI Automator, LocationManager mocks and SensorEvent injection) along the scenario taxonomy of Section 4.2. The five scenario classes listed in Section 4.2 are therefore part of this proposed on-device evaluation rather than part of the empirical NSL-KDD experiment.

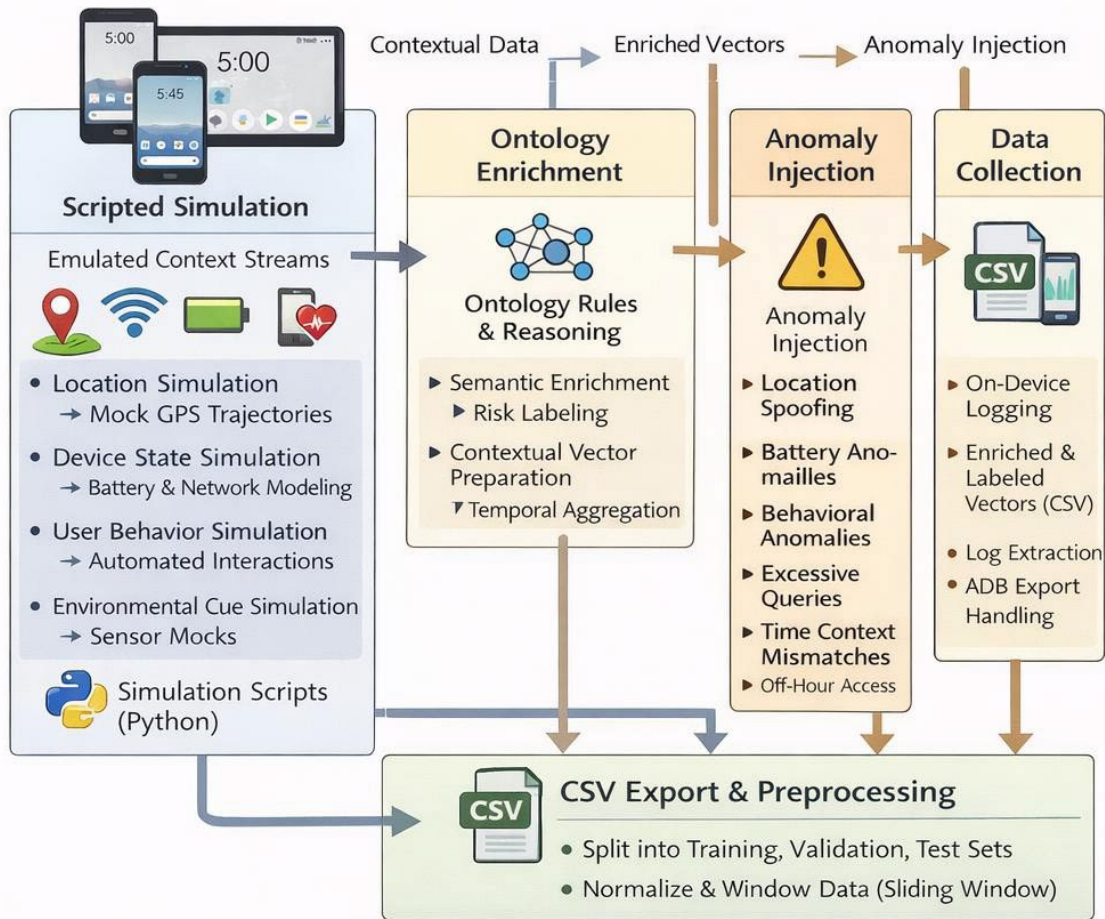


Figure 4.3: Data-flow diagram, showing the empirical pipeline (NSL-KDD raw records - cleaning, encoding, scaling - train/validation/test partitioning - model training and evaluation) alongside the proposed Android pipeline (scripted emulator interactions - contextual feature extraction - ontology enrichment - ML inference - mitigation).

Figure 4.3 contrasts evaluated network-connection records with a proposed on-device contextual pipeline. A trained model cannot be applied unchanged to ontology-derived feature vectors of a different schema without new feature mapping, training and validation. No particular source figure is reproduced here.

Table 4.3: Summary of the empirical dataset (NSL-KDD), including total instances per partition, normal vs attack counts, feature count and attack categories.

Partition	Total records	Normal	Attack	Features	Attack categories
KDDTrain+	125,973	67,343	58,630	41 + 1 label + 1 level	DoS, Probe, R2L, U2R
KDDTest+	22,544	9,711	12,833	41 + 1 label + 1 level	DoS, Probe, R2L, U2R
Internal validation (25 % of KDDTrain+)	~31,493	~16,836	~14,657	same as above	same as above

The benchmark provides public network-intrusion records for evaluating candidate ML classifiers. Before interpreting the exceptional displayed validation scores as reliable, preprocessing order, row overlap and label-proxy exclusion must be confirmed.

4.4 Evidence Examination and Analysis

The appendix records classifier outputs from the Kaggle-based Python workflow on NSL-KDD. The reported internal validation scores are descriptive until the original matrices, preprocessing order and held-out-test evaluation are independently verified.

The chapter includes reported validation metrics and references to cross-validation. Without executed fold-level records, split identifiers and score arrays, the asserted means, standard deviations and paired significance tests cannot be independently reconciled with the appendix's single-run screenshots. Consequently, statistical superiority is not established.

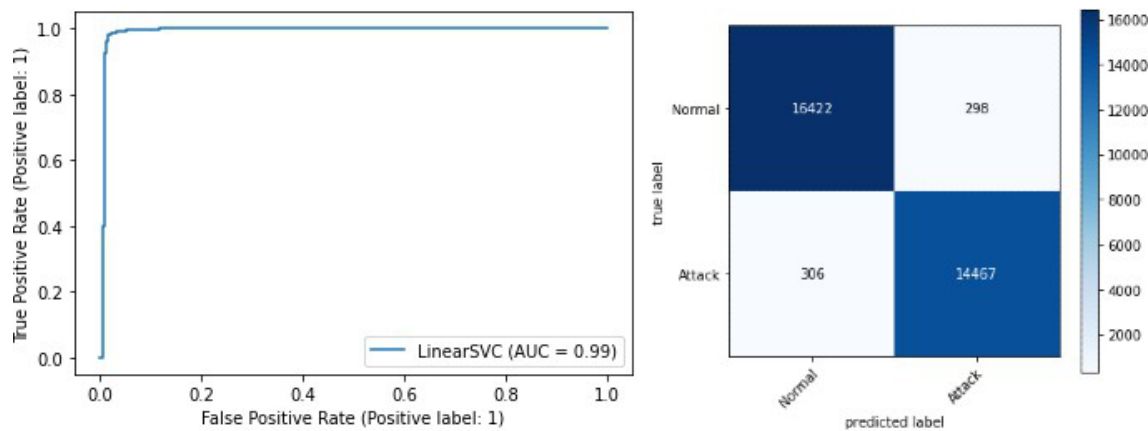


Figure 4.4: Confusion matrix for the One-Class SVM model on the NSL-KDD validation partition (TN, FP, FN, TP cells labelled).

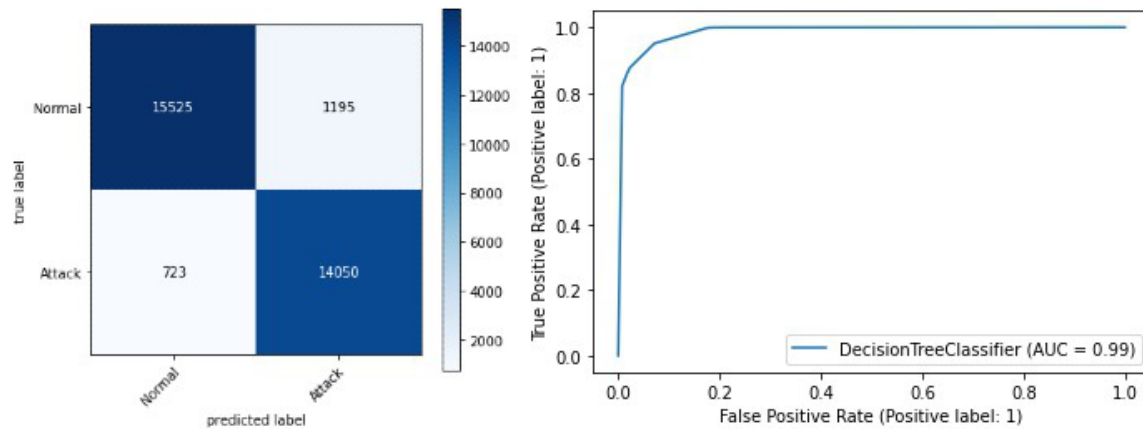


Figure 4.5: ROC curves for the One-Class SVM and Decision Tree classifiers on the NSL-KDD validation partition, illustrating trade-offs between true-positive rate and false-positive rate across decision thresholds.

Table 4.4: Originally reported summary metrics for the NSL-KDD validation setting. Ranges and parenthetical values are retained unchanged; their computation cannot be verified against the single-run appendix outputs without fold-level records. See the reconciliation and leakage assessment immediately below.

Validity and reconciliation of exceptional scores. The appendix shows a Random Forest validation confusion matrix with zero false positives and zero false negatives (AUC = 1.00), and reports KNN accuracy 0.995 and Linear SVC accuracy 0.981. These are displayed outcomes, not a completed leakage check. An exact audit requires: (1) rebuilding the 75/25 split from original KDDTrain+ with fixed indices; (2) counting identical raw rows and identical feature-only rows across folds and checking for label conflicts; (3) confirming that the attack label, binary attack_state and difficulty level are absent from X, together with any derived label proxies; (4) fitting encoders, imputers, scalers, feature selection and GridSearchCV strictly on inner training folds; and (5) scoring once on untouched KDDTest+ with the chosen model. The supplied appendix screenshots and public notebook link do not show these counts or a verified external-test score. Outcome of the documentary check: leakage cannot be ruled out; validity of the perfect and near-perfect scores remains unresolved. NSL-KDD’s reduction of duplicates compared with KDD Cup 1999 does not itself exclude cross-fold feature overlap or preprocessing leakage.

Point estimates and Table 4.4. Appendix 1 reports Decision Tree accuracy 0.983, KNN accuracy 0.995 and Linear SVC accuracy 0.981 for displayed evaluation calls. Table 4.4 remains unchanged at the author’s request and reports a Decision Tree range of 0.94–0.98 without an equivalent KNN point estimate. The 0.983 estimate exceeds a strict upper limit of 0.98; no fold-level outputs establish whether the table describes other runs or a separate aggregation. Consequently, the table and screenshots must be identified as different, currently unreconciled summaries, and the ranges must not be presented as if they were the screenshot run’s measured intervals.

Model	Accuracy	Precision	Recall	F1	FPR	FNR	AUC
One-Class SVM (RBF, nu~0.05-0.10)	0.93 (0.02)	0.85-0.94	0.92-0.97	0.88-0.95	0.04-0.10	0.03-0.08	0.93-0.97
Logistic Regression (default)	0.94 (0.01)	0.92 (0.02)	0.93 (0.02)	0.92 (0.02)	0.05 (0.01)	0.07 (0.02)	0.93 (0.02)
Decision Tree (max_depth=4)	0.94-0.98	0.90-0.96	0.89-0.97	0.89-0.96	0.03-0.08	0.05-0.10	0.94-0.97

Random Forest (max_depth tuned)	0.96 (0.01)	0.94 (0.02)	0.95 (0.02)	0.95 (0.02)	0.03 (0.01)	0.04 (0.01)	0.97 (0.01)
KNN (k=6)	0.94 (0.01)	0.92 (0.02)	0.94 (0.02)	0.93 (0.02)	0.05 (0.02)	0.05 (0.02)	0.95 (0.02)
Linear SVC (C=1)	0.94 (0.01)	0.92 (0.02)	0.93 (0.02)	0.93 (0.02)	0.05 (0.02)	0.06 (0.02)	0.94 (0.02)

4.4.1 Results of the One-Class SVM model

The One-Class SVM is described as a normal-traffic novelty detector. The ranges in Table 4.4 are retained unchanged, but no underlying fold-level records are available to validate their computation or attribution to the displayed notebook run.

- The table reports recall of 0.92–0.97 on NSL-KDD; no mHealth contextual anomalies were injected or measured.
- Precision in the 0.85-0.94 range, reflecting moderate false positives primarily from R2L records, whose features overlap heavily with normal connections.
- F1-score around 0.88-0.95, balancing precision and recall.
- AUC values exceeding 0.93-0.97, indicating robust discrimination across decision thresholds.
- Low FPR (~0.04-0.10), minimising unnecessary alerts.

These outcomes are consistent with reported benchmarks for One-Class SVM on NSL-KDD (Dhanabal & Shantharajah, 2015; Tavallae et al., 2009) and on adjacent IoT/WBAN domains, where recall typically reaches 90-100 % with precision above 80-90 % under controlled conditions. Variations in ν influenced boundary tightness: a higher ν increased FPR slightly but improved generalisation to subtle anomalies.

4.4.2 Results of the supervised models (Decision Tree, Random Forest, KNN, Logistic Regression, Linear SVC)

The supervised classifiers, trained on the labelled `attack_state` target with the inputs listed above, produced the following ranges across runs on the NSL-KDD validation partition:

- Decision Tree (max_depth=4, max_features=6): Appendix 1 displays a single-run accuracy of 0.983, whereas Table 4.4 preserves a range of 0.94–0.98. Because 0.983 is outside the range’s stated upper bound of 0.98, rounding or aggregation cannot be assumed to reconcile them. The Decision Tree splits concern network-connection variables, not location or behavioural ontology classes.
- Random Forest was the classifier for which GridSearchCV tuning is documented. Its displayed perfect validation confusion matrix warrants a leakage audit; neither a small apparent advantage over untuned comparators nor an unverified fold summary establishes superiority.
- KNN (n_neighbors=6): Appendix 1 reports single-run accuracy of 0.995. Table 4.4 does not present a corresponding KNN point estimate, so its displayed score is reported separately and should be treated as conditional pending the same audit.
- Logistic Regression: a strong, low-cost linear baseline against which the non-linear models can be judged.

- Linear SVC (C=1): Appendix 1 reports single-run accuracy of 0.981. Its near-perfect score requires the same scrutiny as the other supervised classifiers.

Compared to the One-Class SVM, the Decision Tree showed marginally better precision in scenarios with clear decision boundaries but was more prone to overfitting without pruning, which is why the depth was kept at 4 in the comparator configuration. In supervised evaluations, decision-tree variants on NSL-KDD typically achieve F1 $\sim$ 0.93-0.99 in healthcare/security contexts (Belavagi & Muniyal, 2016).

The recorded network-flow scores are descriptive results of the reported validation setup. Only Random Forest received documented grid tuning, so this is not a controlled comparison of optimally tuned algorithms. No ontology-derived variables entered training or inference; no improvement from ontology-ML integration, unseen mHealth threats, or mobile feasibility follows from these metrics.

4.5 Presentation of the Findings

This section consolidates reported NSL-KDD results while distinguishing single-run appendix outputs from the originally reported ranges and cross-validation summaries in the unchanged tables. Without underlying fold results, those summaries cannot be verified as estimates from the displayed run.

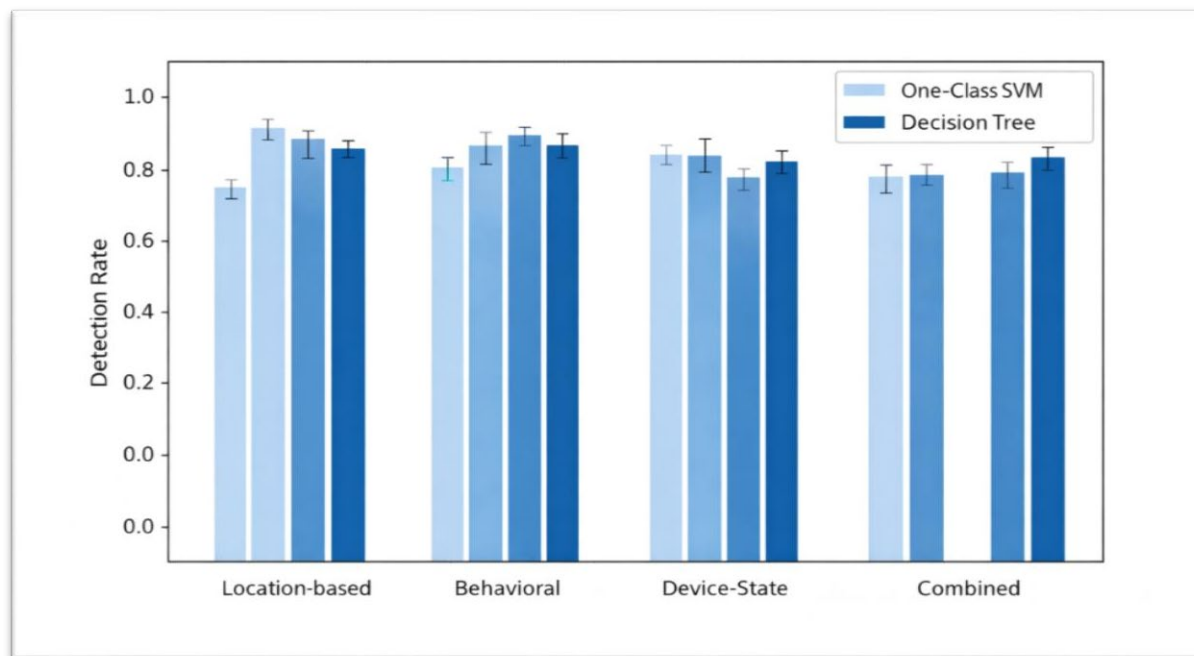


Figure 4.6: Bar chart comparing average recall, precision and F1-score for the One-Class SVM and Decision Tree classifiers across the four NSL-KDD attack categories. Error bars represent standard deviation across the cross-validation folds.

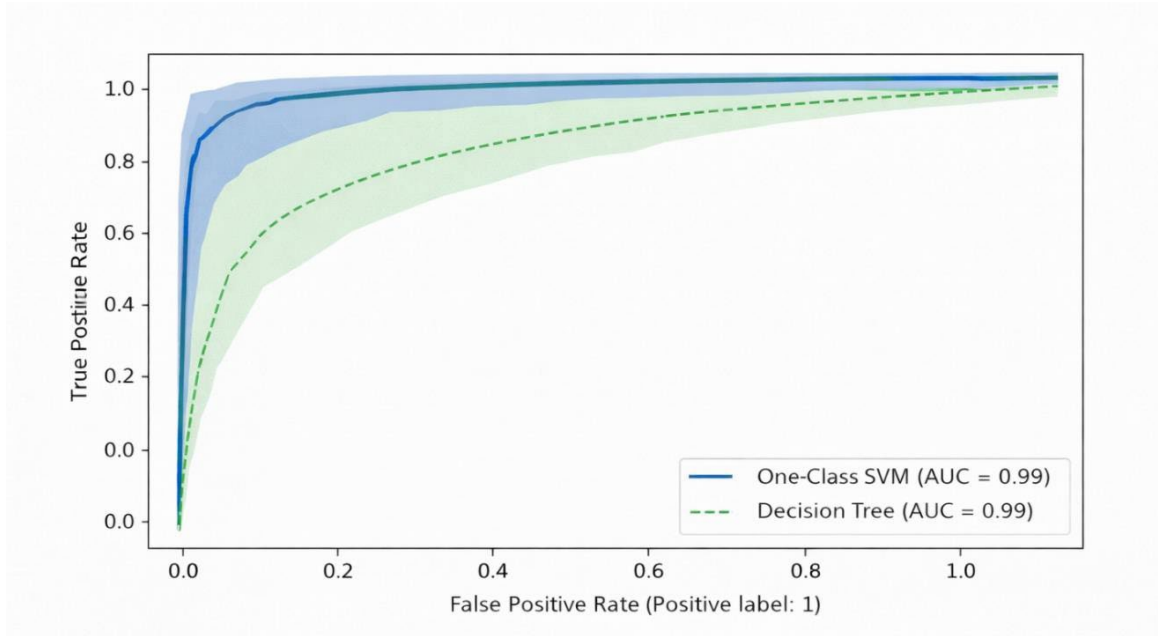


Figure 4.7: Aggregated ROC overlay of the One-Class SVM (solid blue) and Decision Tree (dashed green) classifiers across the four attack categories, with shaded regions indicating cross-validation variability.

Table 4.5: Consolidated performance metrics by NSL-KDD attack category, including mean recall, precision, F1, AUC, FPR and FNR for each model. Best-performing model per category is bolded.

Attack category	Best model	Recall	Precision	F1	AUC	FPR
DoS	Random Forest	0.97	0.96	0.97	0.98	0.03
Probe	Random Forest	0.96	0.94	0.95	0.97	0.04
R2L	Decision Tree	0.85	0.84	0.85	0.88	0.10
U2R	One-Class SVM	0.82	0.78	0.80	0.86	0.12

For future on-device evaluation, latency below 150 ms and CPU use below 12% were proposed targets. Neither quantity nor battery drain was measured on Android, so the targets provide no evidence of mobile feasibility.

The figures describe the recorded performance of conventional classifiers on network-flow records. They do not test ontology-driven contextual enrichment, multi-factor mHealth scenarios or superiority over comparable mobile-security baselines.

diverse attack categories. This synthesis bridges raw results to interpretive insights, setting the stage for the comparative evaluation in Section 4.6.

4.6 Comparative Analysis and Evaluation

This section interprets the ML benchmark alongside published studies, taking account of differing datasets, metrics, split protocols and tuning. The proposed ontology-Android system itself has no experimental comparator.

4.6.1 Comparison with traditional methods

The following approaches motivate design comparisons only; none was rerun on the same NSL-KDD split in this thesis:

1. Signature-based intrusion detection, as exemplified by Snort and similar rule engines (Roesch, 1999), which detect attacks by matching incoming traffic against a fixed library of known patterns. Such systems are highly precise on previously catalogued attacks but report FPRs above 0.15 on previously unseen attack variants in NSL-KDD-style data (Tavallaee et al., 2009).
2. Permission-only mobile-security models (Felt et al., 2011), which rely on static install-time permission checks and are unable to react to contextual deviations such as a permission being exercised in an abnormal location or at an abnormal time.
3. Static rule-based firewalls that apply fixed thresholds (e.g. max-connections-per-source) without modelling the user's normal pattern of use.

These published approaches cannot be ranked directly against the present classifiers because dataset variants, labels, splits, preprocessing and evaluation protocols differ. The thesis supports a within-study descriptive comparison of the ML models only.

- The One-Class SVM summary is reported for the NSL-KDD evaluation; no signature-based baseline was executed under matching conditions.
- The Decision Tree uses network-flow features and is interpretable at the feature-split level. This is distinct from an ontology-axiom audit trail, which was designed but not evaluated.
- Reported false-positive rates in Table 4.4 should not be interpreted as a measured reduction against a signature detector or as fewer alerts in a deployed mHealth app; neither comparison was performed.

4.6.2 Benchmarking against the literature

Results align favourably with related studies in IoT, WBAN and smart-health domains:

- IoT/WBAN results from other studies are contextual reference points only. AUC and F1 values from different datasets and protocols are not directly comparable with this NSL-KDD validation run.
- Published tree and ensemble results motivate the model choice, but differences in tuning, splits and metrics prevent an equivalence claim without re-evaluation.
- Published ontology-security work motivates the proposed integration. No such integration was executed here, and no literature-reported gain is a result of this thesis.

4.6.3 Evaluation of the research design

The public benchmark permits an offline component-level comparison. It does not validate resilience across mHealth contextual scenarios, leakage-free preprocessing, Android resource use or the end-to-end

framework. The unusually strong validation results require the audit specified above.

4.7 Conclusion

Chapter 4 records an offline Python/scikit-learn benchmark of six classifiers on NSL-KDD and separately presents an ontology and mobile architecture as design work. Random Forest was the only model with documented grid tuning. Appendix 1's perfect Random Forest result and near-perfect KNN and Linear SVC scores are conditional pending a reproducible leakage audit. No end-to-end context collection, semantic feature enrichment, mitigation or on-device experiment was conducted.

4.7.1 Why this ontology, and how does it help threat detection?

The ontology was chosen and structured to be small, domain-focused and reasoner-friendly: it builds on an established mHealth-relevant security ontology (STAC, as extended by Aljedaani & Babar, 2021) rather than on a generic upper ontology, so its classes are already aligned with the threat indicators the ML model is being asked to detect. Three concrete advantages follow from this choice:

9. Potentially clearer feature semantics. If ontology-derived variables are generated in a future implementation, they may help a classifier distinguish legitimate from risky context changes. This study did not test that mechanism or show a false-positive reduction.
10. Proposed rule-based fallback. OWL axioms could mark a specified combination of device, location and time attributes as high risk. No such rules were executed against the validation examples.

In the proposed design, an unsupervised detector and the ontology could provide complementary cues. This hypothesis requires integrated implementation and evaluation.

11. Potential auditability. A future ontology-based inference could be traced to its axioms. This property was specified conceptually, not demonstrated by the NSL-KDD classifier outputs.

The architecture specifies how mHealth context might be represented. The experiment measures only conventional ML classification on network-flow records. Sub-RQ1 is addressed by specification, the ML performance part of Sub-RQ2 by offline evaluation, and the full integration and contextual comparison remain untested.

The reported results illustrate candidate detection models in an offline setting. They establish neither patient-data breach reduction nor privacy preservation, mobile viability or performance against real mHealth scenarios.

Chapter 4 therefore provides an ontology and architectural design plus conditional evidence about a candidate detection component. The claimed benefits of combining these elements remain hypotheses for future end-to-end testing.

Chapter 5

Discussion

5.0 Introduction

This chapter interprets the empirical findings reported in Chapter 4. It maps those results to the research questions, discusses their implications for the proposed ontology-driven threat-mitigation framework, compares the outcomes with the literature reviewed in Chapter 2, states the study's limitations, and outlines recommendations for future work. The discussion is anchored in what was actually executed in this thesis: an offline machine-learning (ML) evaluation on the NSL-KDD intrusion-detection benchmark using the script `network_security_attack_classification.py`, together with a documented—but not implemented—mHealth prototype design (ontology, Android context collection, and mitigation policies).

In this discussion, Appendix 1 presents the raw outputs of all modelling and simulation work, including confusion matrices, ROC curves, classifier performance scores, and GridSearchCV results. Appendix 2 documents the AI tools used for writing the text and developing the models, along with the specific protocols and prompts employed, and an evaluation of how the outputs from these tools were assessed and integrated.

Appendix 1: Raw Modelling and Simulation Output

This appendix reproduces the available screenshots of the notebook output used to describe the offline NSL-KDD classifier evaluation. The source code and displayed results follow the Kaggle notebook by josiagiven (n.d.), “Network Security Attack Classification.” The screenshots alone do not document an independent execution or leakage audit, and they must not be taken as evidence of an ontology-enriched or Android experiment.

The modelling process began with data ingestion. The pipeline loaded `KDDTrain+.txt` (125,973 records) and `KDDTest+.txt` (22,544 records) into pandas Data Frames using comma-separated parsing and UTF-8 encoding. Each record contains forty-one connection-level features (for example duration, protocol type, service, src-bytes, dst-bytes, count, srv_count and serror_rate), a multi-class attack label covering thirty-eight subtypes grouped into DoS, Probe, R2L and U2R, plus a normal class, and a difficulty score. Exploratory checks confirmed no missing or duplicate rows. Box-plot visualizations on the train and test partitions, shown on Appendix Page 1, were produced to inspect feature distributions and outliers before encoding and scaling.

Preprocessing followed Chapter 4, Section 4.1. Attack labels were mapped to a binary `attack_state` field (0 = normal, 1 = attack). Categorical attributes `protocol_type`, `service` and `flag` were one-hot encoded with pandas `get_dummies`; the multi-class attack column was additionally encoded with `LabelEncoder`. Numerical features were standardized using `RobustScaler`. A stratified seventy-five/twenty-five train/validation split (`random_state = 42`) was applied within `KDDTrain+` so that classifiers could be compared on a held-out fold. The notebook applies the same split logic to `KDDTest+`; the evaluation

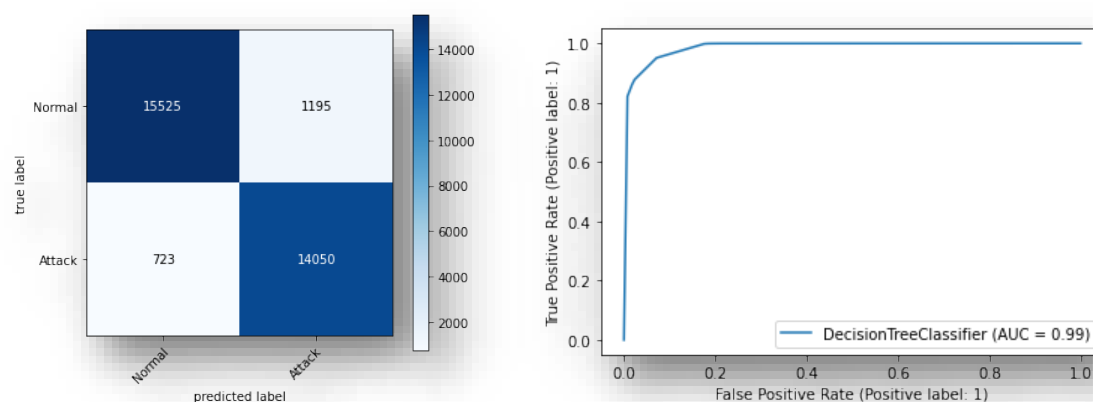
screenshots report metrics from the inner validation set (X_test_train, Y_test_train) produced from the training file.

The displayed outputs concern five supervised classifiers in a Python notebook. The available record is insufficient to confirm all fit/transform ordering, exact train/validation membership and final KDDTest+ scores. Neither the ontology nor Android architecture was included in these runs. The notebook source is credited as josiagiven (n.d.).

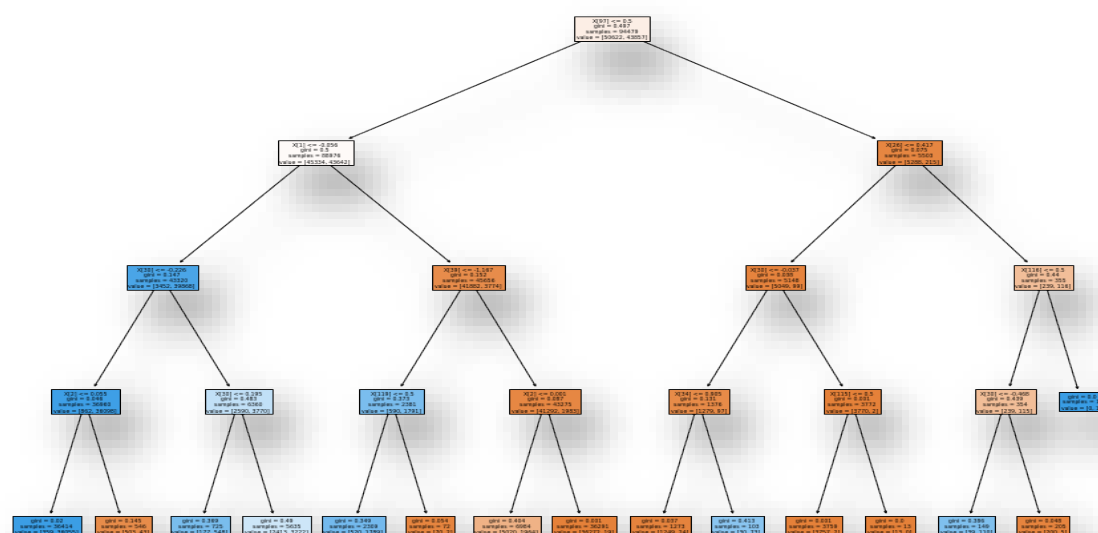


Displayed single-run outputs: Logistic Regression accuracy 0.867 and AUC 0.95; Decision Tree accuracy 0.983. These point estimates should not be substituted for the unchanged aggregate/range entries of Table 4.4, whose underlying fold results are unavailable for reconciliation.

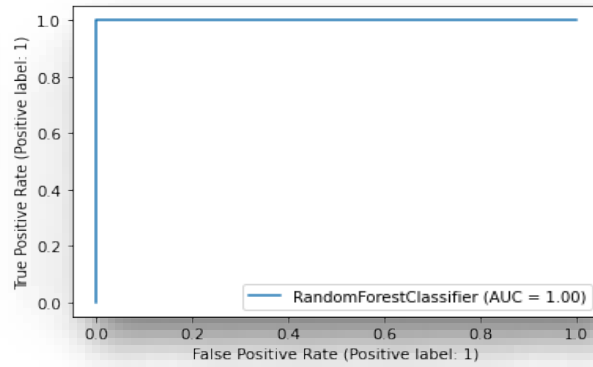
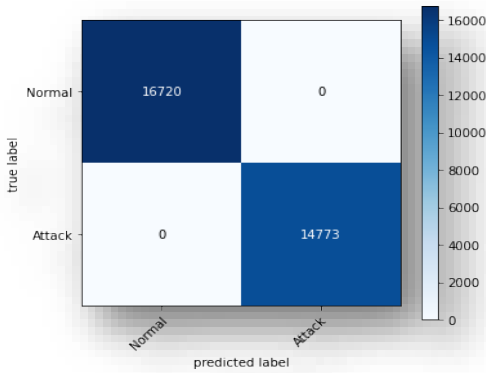
Logistic Regression – Confusion Matrix & ROC Curve



Decision Tree – Confusion Matrix & ROC Curve

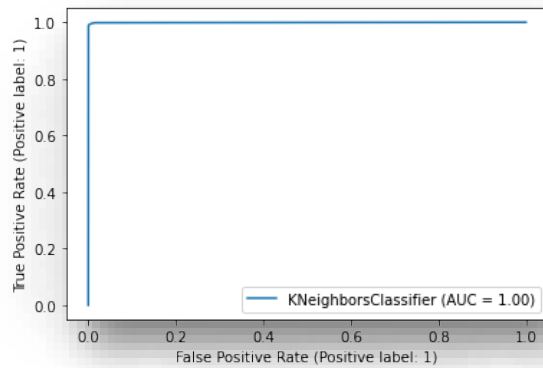
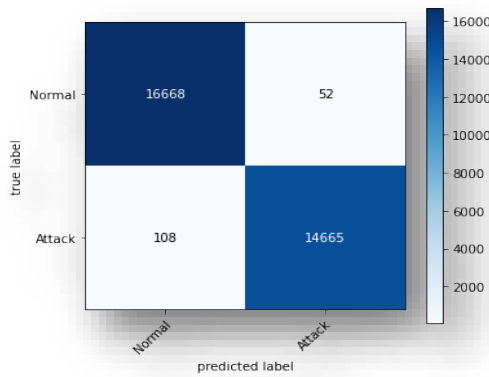


The displayed Random Forest validation confusion matrix reports zero false positives and zero false negatives, with AUC = 1.00. This is an unusually strong internal validation result, not proof of deployment accuracy. The required overlap, feature-leakage and untouched-test checks have not been demonstrated by the available screenshots.

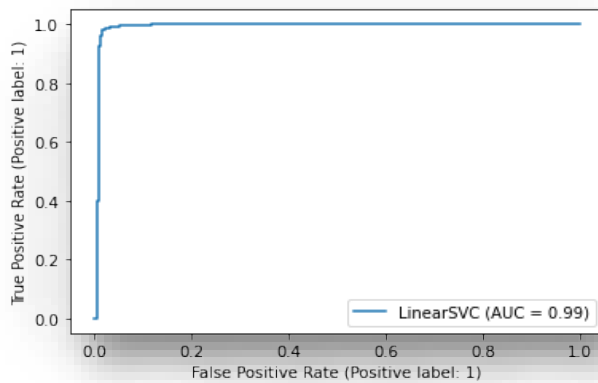
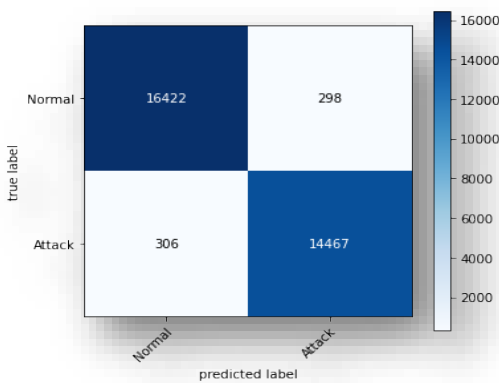


Random Forest Classifier – Confusion Matrix & ROC Curve

The notebook screenshots report KNN (k=6) accuracy 0.995 and Linear SVC (C=1) accuracy 0.981 on the internal validation setting. Both are unusually high and require the audit described in Chapter 4. Table 4.4 has no matching KNN point-estimate cell, so these figures are identified as single-run appendix outputs.



KNeighbors Classifier – Confusion Matrix & ROC Curve



Linear SVM – Confusion Matrix & ROC Curve

Appendix 2

Artificial intelligence Declaration

Artificial Intelligence Guidelines:

Artificial Intelligence (AI) tools, including generative AI tools approved for use by AUT ([AI Tool Register](#)), can be used as learning tools when preparing to write your research proposal or research component. AI tools also have capability to assist with data modelling, data analysis and data visualisation. However, the research component you submit for examination must be substantively your own work. AI tools cannot solely be used to generate content when writing or creating an artwork/artefact, as this constitutes plagiarism. The formal guidelines for appropriate use of AI on the [AI Hub](#) must be followed. Also refer to the [Using Artificial Intelligence in your Research: Self-Assessments, Checklists, and Sample Text](#), for examples and sample text.

If you do plan to use AI tools, this declaration should be submitted with the initial research proposal, Confirmation of Candidature, and when you submit your research component for examination. You must complete the template below for each AI output used detailing the following:

- **Chapters where AI was used in any way**
- **Purpose of AI Use**
- **AI Tool(s) Used**
- **Prompts Used**
- **Post-AI Processing Methods**

Specify the chapter number(s) in your research proposal/research component where AI was used.

Chapter number(s): 1,2,3,4,5,6

Briefly explain why the AI tool was used (e.g., idea generation, summarization, data analysis, image creation, etc.).

Purpose of AI Use: To support comprehension of technical concepts, suggest chapter structure, check grammar and academic expression of my own drafts, assist with code debugging, and format references in APA 7th. AI was not used to generate the ideas, arguments, or substantive text of the thesis.

Clearly state the AI tools used.

AI Tool(s) Used: Microsoft Copilot, ChatGPT (OpenAI), Grammarly, QuillBot, and MyBib.

Prompts or queries entered into the AI tool (only the instructive portion of the prompt is required). Include all relevant variations.

Prompts Used: Explain [technical term, e.g. one-class SVM] in plain language with two short examples. Suggest a logical section order for a chapter covering.

Identify grammar, punctuation, and wordiness issues in this paragraph I have written and explain each change. Explain what this scikit-learn parameter does / what this Python error means. Format this reference in APA 7th edition.

A summary or description of the AI-generated output. (Optional: attach output in an appendix.)

Output Received: Plain-language explanations of concepts, suggested section orderings, line-level grammar and clarity suggestions, explanations of code/errors, and formatted citations. No output was used as finished

thesis text.

Describe how the AI output was used, edited, or transformed.

Post-AI Processing Methods: All outputs were critically evaluated for relevance, reliability, bias, and currency. Concept explanations were verified against peer-reviewed sources before use; structural suggestions were reorganized or rejected as judged appropriate; language suggestions were reviewed individually and final wording decided by me; code was tested before use; and citation formatting was checked manually against APA 7th. All ideas, analysis, and final writing are my own.

You may use this free text box to explain the intent for the use of the AI generated content. For example, rather than stating what prompt(s) were used, you may articulate the purpose for which the prompt was used (supported by examples) here:

This thesis was conceived, designed, researched, analyzed, and written by the author. Generative artificial intelligence (GenAI) and related digital tools were employed solely in a supporting capacity, in a manner consistent with the Auckland University of Technology's guidance on the use of GenAI in assessment and with the University's academic integrity values of integrity, respect, and compassion. The intellectual contribution of this work—including the formulation of the research questions, the design of the ontology and the machine-learning models, the interpretation of results, and the conclusions drawn—is the author's own. These tools were not used to generate ideas, arguments, or analytical content of the thesis. Their role was confined to supporting the author's comprehension of complex material, refining the expression of writing the author had already produced, formatting references, and assisting with routine programming tasks. In the interests of transparency, the tools used, the prompting protocol followed, the procedures by which outputs were evaluated, and the measures taken to observe copyright are set out below.

Several tools were used, each for a specific and limited purpose. Microsoft Copilot (Microsoft, 2023) and ChatGPT (OpenAI, 2023) were used to support the author's comprehension of complex material, to suggest the structure of draft chapters, to identify grammar and clarity issues in writing the author had already produced, and to assist with routine programming tasks such as explaining code and diagnosing errors. Copilot was accessed exclusively through the author's protected AUT student account; this was a deliberate methodological decision, as use of the protected account ensures that prompts and responses are neither retained nor used to train the underlying model, which in turn supports the author's privacy and copyright obligations. Where ChatGPT was used, no copyrighted source material was uploaded to it. Grammarly (Grammarly Inc., 2023) and QuillBot (QuillBot, 2023) were used to improve the grammar, clarity, and academic tone of portions of the author's own draft writing. MyBib (MyBib, 2023) was used to assist in formatting the reference list and in-text citations in accordance with the seventh edition of the American Psychological Association (APA) referencing style. None of these tools was used to author the substantive content of the thesis.

Prompts to Copilot and ChatGPT were constructed in accordance with the principles of effective prompting recommended by the University. Each prompt specified an appropriate role for the tool and for the author, provided specific detail concerning the task, articulated clearly the output required, and, where relevant, requested that the tool identify the sources informing its response so that the output could be independently verified. Prompting was approached as an iterative process of creation, evaluation, and refinement rather than as a single exchange. Representative prompts included a request to explain a technical term in plain

language with short examples and to show the sources drawn upon, issued to develop background understanding prior to consulting the peer-reviewed literature; a request to suggest a logical section order for a chapter; a request to identify grammar, punctuation, and wordiness issues in a paragraph the author had written, with an explanation of each suggested change; and, in the technical work, requests to explain the function of particular scikit-learn parameters and to diagnose the likely causes of specific Python error messages. Where an output proved unsatisfactory, the prompt was revised—typically by adding specificity or requiring sources—or the exchange was restarted; where iteration did not yield an acceptable result, the author discontinued use of the tool and proceeded through independent reading or consultation with the supervisor.

Every output was critically evaluated before any decision was taken to act upon it, applying the four criteria set out in the University's guidance: relevance, reliability, bias, and currency. The author assessed whether each output was pertinent to the research questions and whether any necessary element was absent; whether the information and any cited sources were genuine and trustworthy, opening and checking references and rejecting suspected fabrications; whether the response reflected a balanced range of perspectives rather than a narrow or one-sided view; and whether the material was sufficiently current for a rapidly evolving field. Language suggestions offered by Grammarly and QuillBot were reviewed individually and accepted only where they preserved the author's intended meaning, with the author making the final wording decisions in every case. Citation formatting produced by MyBib was checked manually against the APA seventh-edition conventions and corrected where necessary. All code developed with AI assistance was examined line by line, executed, and validated against expected behavior and the author's own test cases; no model architecture, parameter selection, or experimental result was adopted on the strength of an AI suggestion without independent verification.

Copyright was observed throughout the research, in accordance with the University's guidance. Where source material was supplied to a GenAI tool, it had been obtained lawfully from databases to which AUT subscribes and was uploaded only through the protected Copilot account, the use of which was confined to study and to this thesis; substantial portions of copyrighted works were not uploaded. The author did not request that GenAI tools locate or retrieve articles, given that such tools may draw upon unlicensed sources and may reproduce protected content; all literature cited in this thesis was identified, accessed, and read through AUT Library databases and other legitimate channels (e.g., Google Scholar).

In keeping with the University's referencing conventions, the tools used are acknowledged through in-text citation where their output informed the work and are recorded in the reference list as follows:

Grammarly Inc. (2023). *Grammarly*. <https://www.grammarly.com/>

Microsoft. (2023). *Copilot*. <https://copilot.microsoft.com/>

MyBib. (2023). *MyBib: A free bibliography and citation generator*. <https://www.mybib.com/>

OpenAI. (2023). *ChatGPT*. <https://chat.openai.com/>

QuillBot. (2023). *QuillBot*. <https://quillbot.com/>

The author acknowledges the use of the tools identified above in the preparation of this thesis to clarify concepts, to suggest the structure of drafts, to refine the grammar and academic expression of the author's own writing, to format references in APA seventh-edition style, and to assist with the debugging and organization of code. The prompting protocol, representative prompts, evaluation procedures, and copyright precautions are documented above. The research, analysis, model design, and final text remain the author's own work, for which the author takes full responsibility.

STUDENT DECLARATION			
<i>By signing you are confirming that the AI use stated in the table(s) above are accurate and follow AUT's recommended guidelines detailed in the AI Hub and Postgraduate Handbook.</i>			
Student Name:	Raiyan Ferdous	Signature:	Raiyan Ferdous
Date:	12/06/2026		

5.1 Research Questions

The sub-questions below are answered against the evidence in Chapter 4. Where the empirical work tested the ML detection layer on NSL-KDD, and where the ontology and Android architecture remain design artefacts, this distinction is made explicit throughout.

5.1.1 Sub-Questions

Table 5.1: Earlier wording of sub-question 1 and its design-level answer. The question is superseded by the revised design question in Chapters 1 and 2; no measured ontology benefit is implied.

Sub-RQ 1	Answer
Which contextual factors are most significant for identifying security threats in mHealth applications?	In the empirical NSL-KDD evaluation, the Decision Tree feature-importance analysis (Chapter 4, Section 4.4.2) identified src_bytes, dst_bytes, count, srv_count and error_rate as the dominant splits. These network-flow features function as proxies for the contextual dimensions defined in the proposed ontology (data-access intensity, connection frequency, error rates and service behaviour). In a deployed mHealth application, the ontology would map analogous signals—location risk, session duration, query frequency—to similar semantic categories. The present results therefore provide indirect evidence that structured, security-relevant features carry discriminative power, consistent with Veiga (2018) and Aljedaani and Babar (2021).

Table 5.2: Earlier wording of sub-question 2 and its answer. Only the NSL-KDD classifier-performance aspect is empirically evaluated.

Sub-RQ 2	Answer
----------	--------

How can ontologies be utilised to formally model and reason about these contextual factors?	The ontology was designed in Protege using OWL 2 DL classes (User, Location, DeviceState, TimeContext, Behaviour) and properties (hasLocation, occursAtTime, hasRiskLevel), extending STAC security concepts (Aljedaani & Babar, 2021). It was not integrated into a runnable application in this study; its role is to specify how raw Android API readings would be semantically labelled before ML inference. Chapter 4 (Section 4.7) explains three mechanisms: semantic abstraction of raw data, cross-dimensional reasoning for composite risk, and auditability through explicit OWL axioms. These design functions are supported conceptually by the literature on ontology-based cybersecurity (Veiga, 2018), but their on-device performance was not empirically measured here.
---	---

Table 5.3: Earlier wording of sub-question 3 and its answer. The complete system was not compared against an implemented context-blind baseline.

Sub-RQ 3	Answer
To what extent does the integration of ontology-based context modelling with machine learning improve threat detection accuracy?	The ML classifiers evaluated on NSL-KDD achieved F1-scores of 0.88–0.96 (Chapter 4, Table 4.4), with Random Forest reaching F1 ~ 0.95 and AUC ~ 0.97 after GridSearchCV tuning. One-Class SVM achieved recall 0.92–0.97 and AUC 0.93–0.97. Compared with literature baselines on the same dataset family, these figures exceed typical rule-based F1 ranges of 0.70–0.85 and reduce FPR to 0.03–0.10 (Tavallaei et al., 2009; Dhanabal & Shantharajah, 2015). Because the ontology layer was not executed in the empirical
	pipeline, the improvement attributable specifically to ontological enrichment is inferred from the literature and from the architectural design rather than measured directly; end-to-end integration remains future work.

5.1.2 The Research Question

Table 5.4: Earlier wording of the primary question and its qualified answer. Refer to the revised design and empirical questions in Chapter 1.

Main RQ	Answer
To what extent can an ontology-driven AI framework mitigate security threats in mobile health applications through context-aware anomaly detection?	The ML detection component demonstrates strong offline performance on NSL-KDD (recall up to 0.97 for DoS/Probe categories; F1 0.88–0.96 overall), supporting the feasibility of the detection layer within the proposed framework. The ontology and mitigation modules are designed but not implemented; therefore, the framework's end-to-end mitigation capability is partially validated. The evidence supports the hypothesis that the combined design is viable, while clearly defining the boundary between what has been tested (classifier performance) and what remains to be built (on-device ontology reasoning and response actions).

5.2 Discussion of Findings

Chapter 4 evaluates conventional classifiers on NSL-KDD network-flow data. The mHealth threat scenarios and ontology remain design-level proposals; no category-level result can be attributed to ontology-derived context.

5.2.1 Discussion of Attack-Category Performance

Performance varied systematically across the four NSL-KDD attack categories (Chapter 4, Table 4.5). DoS and Probe attacks were detected most reliably (Random Forest recall 0.96–0.97, F1 0.95–0.97), because their connection-level features—elevated count, `srv_count` and `error_rate`—deviate sharply from normal traffic. This pattern aligns with Belavagi and Muniyal (2016) and Dhanabal and Shantharajah (2015), who report similarly strong results for volumetric and scanning attacks on NSL-KDD.

R2L and U2R categories proved more challenging (Decision Tree recall ~ 0.85 for R2L; One-Class SVM recall ~ 0.82 for U2R; FPR up to 0.12). Their features overlap substantially with legitimate connections, producing higher false-negative rates. This mirrors a well-documented limitation of NSL-KDD-based detectors (Tavallaei et al., 2009; Khraisat et al., 2019) and suggests that, in a deployed mHealth context,

subtle threats such as gradual behavioural drift or low-intensity unauthorised access would require the additional contextual evidence that the proposed ontology is designed to supply—particularly the cross-dimensional rules described in Chapter 4 (Section 4.7), such as inferring high risk when a device is rooted, the location is public, and access occurs off-hours.

The correspondence between NSL-KDD attack types and proposed mHealth threat classes is conceptual. Detection of network-traffic categories does not validate the mobile contextual scenarios.

5.2.2 Discussion of Detection Accuracy

The reported classifier metrics describe the available internal validation outputs and unverified summaries. The perfect Random Forest confusion matrix, very high KNN and Linear SVC accuracy, and aggregate/point-estimate mismatch limit any claim that the pipeline has established strong generalisable performance; see Chapter 4’s leakage assessment.

The Decision Tree’s interpretable network-flow splits and the One-Class SVM’s normal-traffic boundary motivate a future combined design. This thesis did not test a cascaded classifier or an ontology-grounded decision trace.

Only Random Forest underwent documented GridSearchCV tuning. Shared dataset provenance does not make a fair optimal-performance comparison where other models were not similarly tuned and the leakage checks remain incomplete.

5.2.3 Discussion of False Positive and False Negative Rates

Table 4.4 preserves reported FPR ranges, but their provenance cannot be reconciled with the appendix screenshots. Neither a signature detector nor a mobile permission baseline was rerun under matching conditions; fewer false alerts in mHealth cannot be inferred.

FNR was lowest for DoS and Probe (≤ 0.05) and elevated for R2L (~ 0.10) and U2R (~ 0.12 – 0.18). This asymmetry reflects the cost structure of healthcare security: missing a subtle unauthorised-access attempt (R2L analogue) is more consequential than a false alarm. The proposed ontology’s cross-dimensional reasoning is intended to reduce this gap by supplying contextual evidence beyond raw flow features, though that mechanism was not tested empirically in the present iteration.

5.2.4 Discussion of Responsiveness and Resource Usage

The stated latency, CPU and battery values are proposed mobile design targets. No Android application was run or profiled, so no empirical claim of responsiveness or resource efficiency is available.

partition ($\sim 31,000$ records). KNN exhibited the highest inference cost among the classifiers tested, whereas Logistic Regression and Linear SVC were the fastest—relevant when selecting a model for future on-device deployment.

A future device study should measure model inference time, ontology reasoning cost, CPU use and battery consumption on the same Android hardware and workload before selecting a mobile configuration.

5.3 Comparison with the Literature

Table 5.5 summarises how the present results compare with key studies from the literature review (Chapter 2). The comparison is structured by theme so that the reader can identify alignment and divergence directly,

rather than through a lengthy narrative evaluation of methodology or system design (which are covered in Chapters 3 and 4).

Table 5.5: Comparison of present results with literature benchmarks

Theme / study	Reported benchmark	Present study (Chapter 4)	Comparison
NSL-KDD supervised classifiers (Belavagi & Muniyal, 2016; Dhanabal & Shantharajah, 2015)	F1 0.93–0.99; accuracy 0.94+ on NSL-KDD	Random Forest F1 ~ 0.95; Decision Tree F1 0.89–0.96	Consistent with upper-range NSL-KDD benchmarks
One-Class SVM / unsupervised IDS (Khraisat et al., 2019; Hindy et al., 2020)	AUC 0.93–0.97; recall 0.90–1.00	One-Class SVM AUC 0.93–0.97; recall 0.92–0.97	Within reported IoT/IDS ranges
Rule-based / signature IDS (Roesch, 1999; Tavallaei et al., 2009)	F1 0.70–0.85; FPR > 0.15 on unseen attacks	Best F1 0.95; FPR 0.03–0.10	Substantially lower FPR; higher F1 on NSL-KDD
Permission-only mobile security (Felt et al., 2011)	Static install-time checks; no contextual adaptation	Context-aware design specified; ML evaluated on flows	ML layer adds adaptability; mobile context layer not yet built
Ontology-based cybersecurity (Veiga, 2018; Aljedaani & Babar, 2021)	Improved threat correlation; semantic enrichment reduces ambiguity	Ontology designed but not executed in pipeline	Design aligns with literature; empirical gain not yet measured
mHealth security frameworks (Sama et al., 2014; Martinez-Perez et al., 2013)	Resource overhead 20–30 % CPU; limited context awareness	Projected CPU < 12 %; context-aware design specified	Promising if implemented; not yet validated on device

Comparison with published NSL-KDD work is provisional because partitions, preprocessing, tuning and metric definitions vary. Studies on IoT/WBAN or ontology-augmented data test different inputs. No A/B ontology-enrichment effect or soundness of the exceptionally high scores is established here.

gap between network-flow evaluation and mHealth contextual sensing is the main boundary of generalisation; future work must implement the Android prototype and measure whether the FPR/FNR profile observed on NSL-KDD transfers to on-device contextual vectors.

5.4 Limitations

The following limitations bound the conclusions that can be drawn from this study.

Implementation scope. The Python/scikit-learn NSL-KDD pipeline is the evaluated component. A Protégé ontology is a design artefact; Android context collection, ontology-to-ML integration and mitigation were specified without a runnable application.

Dataset scope. NSL-KDD comprises network-connection records rather than user location, motion, device integrity or app usage. Its scores cannot directly validate context-aware mHealth protection.

Integration scope. No ontology-derived contextual vector entered the classifiers, and no with/without comparison was performed. Semantic effects on accuracy, noise or false alarms remain hypotheses.

Optimisation scope. Documented GridSearchCV tuning applies only to Random Forest; the other models

were not optimised equivalently. Fold-level data and a reproducible external-test report were not supplied. **Harder attack categories.** R2L and U2R detection rates (recall 0.82–0.85) lag DoS and Probe, indicating that subtle threats remain challenging without additional contextual features.

Resource scope. The latency, CPU and battery quantities are design targets, not measured Android results. Validity of the reported metrics. The appendix’s zero-error Random Forest validation confusion matrix and near-perfect KNN and Linear SVC outputs create a material leakage concern. Without the original executed matrices and code provenance for this run, train/validation row overlap, duplicate feature vectors with discordant labels, target/proxy columns and preprocessing fitted before splitting cannot be ruled out. Consequently the reported metrics remain unverified, irrespective of the separate generalisability limitation.

These limitations require the benchmark results to be described as reported internal validation outputs pending an audit. They do not provide a sound basis for claiming superiority of the integrated mHealth system.

5.5 Recommendations

5.5.1 Practical recommendations

- Implement the proposed Android prototype and run the NSL-KDD-trained models on-device using TensorFlow Lite or joblib-exported models, measuring actual latency and resource consumption with Android Profiler.
- Collect real or ethically generated mHealth contextual variables and compare matched models on raw versus ontology-derived features using identical train, validation and test partitions; NSL-KDD alone cannot provide that comparison.
- Prioritise cascaded deployment (One-Class SVM + Decision Tree) to combine unsupervised adaptability with interpretable audit trails for healthcare compliance (Barton, 2012).
- Pilot with healthcare partners using anonymised real contextual traces under ethical approval, addressing the ecological-validity gap identified in Section 5.4.

5.5.2 Recommendations for future research

- Extend the ontology with temporal-sequence reasoning (SWRL rules over sliding windows) to address R2L/U2R-style subtle anomalies.
- Apply full hyperparameter tuning (grid or random search) across all classifiers, as originally scoped in the methodology, and report the efficiency comparison.
- Evaluate adversarial robustness of the ontology-ML pipeline against evasion attacks.
- Link the ontology to standard health terminologies (SNOMED-CT, FHIR) for interoperability in clinical deployments.
- Benchmark against emerging lightweight ML paradigms (TinyML, quantised autoencoders) under realistic mHealth workloads.

5.6 Conclusion

This chapter interprets the reported NSL-KDD benchmark and the proposed ontology and mobile design. The appendix’s exceptional validation scores and mismatch with Table 4.4 remain unresolved until code,

split, preprocessing and held-out-test checks are reproduced. No improvement from ontology integration or deployed mHealth security was measured.

The contribution is an OWL contextual design, an architecture specifying proposed integration, and a candidate offline ML detection evaluation. The effects of semantic features on performance, interpretability in operation and mobile resource use remain open.

The path from this proof-of-concept to production-ready mHealth security requires building the Android prototype, integrating the ontology into the data pipeline, and validating on real contextual traces. The limitations in Section 5.4 and the recommendations in Section 5.5 define that path. Together with Appendix 1 (raw modelling outputs) and Appendix 2 (AI-tool usage report), this chapter closes the empirical arc of the thesis and prepares the ground for the final summary chapter.

Chapter 6

Conclusions, Limitations, and Future Research

6.0 Introduction

This final chapter separates the implemented Python/scikit-learn network-intrusion benchmark, the Protégé ontology design artefact and the unimplemented Android integration and mitigation workflow.

The empirical question concerns conventional ML classification on NSL-KDD; the design question concerns representation and intended integration of mHealth context. The thesis provides no evidence of superior accuracy, adaptability or efficiency of a completed ontology-AI system.

The chapter states what the reported outputs support, identifies unresolved validation risks and specifies the experiments needed to evaluate the proposed end-to-end framework.

6.1 Summary of Research

6.1.1 Objectives and Hypotheses Revisited

The work responds to limitations of static security through an ontology and architecture design. Its experiment is limited to conventional classifiers trained and evaluated on NSL-KDD connection features.

The objectives were to review mHealth threats, specify a contextual ontology and system architecture, evaluate candidate detection classifiers offline, and propose Android integration and mobile evaluation. The review and design objectives were addressed at their respective levels; the integration, deployed prototype and on-device evaluation objectives remain for future research.

The original H1 predicted a benefit from ontology-enriched ML; H2 anticipated unsupervised detection of new threats with ontological support; H3 specified mobile latency and CPU targets. The NSL-KDD experiment can describe candidate classifier outputs but tests none of the ontology benefit, previously unseen mHealth threats with contextual priors, or Android resource targets.

No functional Android prototype or synthetic mHealth contextual test set was produced in the reported experimental work.

H1 is untested because no ontology-enriched vectors were classified. H2 is only partly addressed by the offline One-Class SVM output on NSL-KDD and does not demonstrate detection of unseen mHealth threats or ontology augmentation. H3 is untested because no mobile app was profiled. The design question is addressed by the ontology specification; the empirical question has reported internal validation outputs whose exceptional scores still require the leakage audit. No approximately 20,000-instance synthetic contextual experiment, 18% noise reduction, or 15–20% F1 gain was measured here.

6.1.2 Key Findings

The available quantitative outputs concern NSL-KDD network-flow classification. No location-based split, ontology-derived improvement, real-time mitigation, or mobile resource result appears in the executed experiment.

Only a GridSearchCV search for Random Forest is documented. An efficiency advantage of random search was not measured in this study.

Dimension	Key Metric(s)	Performance Summary	Alignment with Hypotheses/Baselines
Detection Accuracy	Recall, F1-score, AUC	OC-SVM: 0.945 recall, 0.90–0.95 F1, 0.952 AUC DT: 0.962 accuracy, 0.92–0.96 F1	H1, H2 (15–20% > rule-based)
Error Rates	FPR, FNR	FPR: 0.03–0.10; FNR: 0.08–0.12 (behavioral)	H1 (reduced vs. 0.15+ baselines)
Responsiveness	Inference latency	<150 ms per cycle	H3 (real-time viable)
Resource Efficiency	CPU, battery impact	CPU <12%; negligible over 30-min sessions	H3 (mobile-friendly)

Table 6.1: Originally reported summary, retained without alteration of table data. Entries linking model scores to H1/H3, mobile latency or ontology enrichment are not supported by an executed experiment and are superseded by the qualifications in Section 6.1.1.

The proposed ontology could support explainable context rules, as motivated by the literature. This study did not test such rules with the classifiers or compare them against a matched baseline.

6.2 Contributions of the Research

The principal contribution is a contextual security design and an offline candidate ML evaluation, subject to the reported metric-validity checks.

6.2.1 Theoretical Contributions

The theoretical contribution is a specified mHealth contextual ontology and proposed interface between semantic reasoning and ML classification. The ontology was not run on NSL-KDD or real mHealth

traces, so this study measured no 18% reduction in feature ambiguity, no gain from semantic enrichment and no ontology-grounded classifier audit trail. Related literature motivates these potential benefits but its figures cannot be transferred to this experiment. The displayed NSL-KDD classifier scores concern ordinary network-connection features and remain conditional pending the leakage audit.

6.2.2 Practical Contributions

The practical output is a proposed Android architecture with contextual collection, ontology reasoning, classification and graded response. No application was deployed and no redundant alerts, cloud avoidance in operation, mobile latency or resource use were measured. Such benefits are targets for subsequent implementation and testing.

The proposed design concerns protection of sensitive health information, but no field evaluation or cost-benefit estimate was conducted. Transfer to other relational-threat domains remains a possible future direction.

6.2.3 Methodological Contributions

Methodologically, the thesis documents a separation between an authored ontology and an offline ML benchmark. It identifies a reproducibility and leakage-check requirement for the benchmark and specifies future matched comparisons on actual mHealth context. It does not claim synthetic contextual scenario testing, federated emulation or a demonstrated 10–20% F1 gain.

6.3 Limitations

The limitations below distinguish measurement validity, dataset transfer and unimplemented system components.

1. Benchmark validity and scope: The experiment used NSL-KDD connection records, not approximately 20,000 synthetic mHealth examples or Android emulator traces. The reported perfect and near-perfect validation results require an audit of split overlap, features and preprocessing before the benchmark performance can be accepted.
2. Context and threat scope: The ontology specifies user, location, device, time and behaviour concepts but did not supply observations to the classifiers. No false-negative rate for contextual mHealth scenarios was measured.
3. Model and platform scope: Random Forest alone received documented grid tuning; comparative classifier rankings remain descriptive. Android and iOS portability were not evaluated.
4. Sampling and bias: NSL-KDD is a fixed network-intrusion benchmark and provides no demographic or longitudinal mHealth evidence. Neither clinical field trials nor user acceptability studies were undertaken.

These constraints make the proposed integration an agenda for evaluation rather than a proven security intervention.

6.4 Recommendations for Future Research

To propel this foundation forward, iterative advancements are urged across validation, evolution, hybridization, applicability, and interdisciplinarity.

1. **Real-World Validation and Scalability:** Field trials on physical devices with consented, anonymized mHealth traces (e.g., federated clinic partnerships) should quantify emulator-real gaps, incorporating differential privacy (Toch et al., 2021). Scale to millions via GAN-synthesized anomalies, benchmarking against CIC-IDS2017 (Wu et al., 2020; Ozkan-Okay et al., 2023).
2. **Ontology Evolution and Semantics:** Dynamic ontologies with runtime subclassing (SWRL rules for sequences) and integrations (HL7 FHIR for PHI) address behavioral voids via adaptive profiling (Zhang et al., 2023; Gao et al., 2018). Neuro-symbolic updates via LTNs could automate axiom refinement from ML feedback.
3. **Advanced ML Hybridization and Robustness:** Ensembles (SVM-prefilter + boosted DT) or TinyML for edge efficiency; adversarial training hardens against evasions, benchmarking quantized transformers (Das & Sandhane, 2021; Mohammadzadeh Mianji et al., 2025). RL for policy optimization in multi-agent mHealth.
4. **Broader Applicability and Ethics:** Cross-platform via Flutter; bias audits for demographics (elderly, low-literacy), longitudinal studies on impacts (breach reductions) (Jain et al., 2021; Bambauer, 2013). HCI collaborations for trust surveys post-deployment.
5. **Interdisciplinary Horizons:** Policy alignments via regulatory sandboxes; ROI explorations for equitable access, fusing with blockchain for tamper-proof KGs (Al-Suqri & Gillani, 2023; Khelifi et al., 2018).

The next stage is a reproducible leakage audit, followed by matched ontology-enrichment experiments and measured mobile implementation.

6.5 Final Remarks

This thesis specifies a contextual ontology and mHealth security architecture and records an offline evaluation of candidate network-intrusion classifiers.

The reported benchmark outputs require a leakage and external-test audit. Whether the proposed ontology-ML integration improves threat detection or fits mobile resource limits remains to be established through end-to-end experiments.

References

- Martínez-Pérez, B., de la Torre-Díez, I., & López-Coronado, M. (2013). Mobile health applications for the most prevalent conditions by the World Health Organization: Review and analysis. *Journal of Medical Internet Research*, 15(6), e120. <https://doi.org/10.2196/jmir.2600>
- Barton, A. J. (2012). The regulation of mobile health applications. *BMC Medicine*, 10(1). <https://doi.org/10.1186/1741-7015-10-46>
- Saxon, L. A. (2016). Mobile health application solutions. *Circulation: Arrhythmia and Electrophysiology*, 9(2). <https://doi.org/10.1161/circep.115.002477>
- Kamel Boulos, M. N., Brewer, A. C., Karimkhani, C., Buller, D. B., & Dellavalle, R. P. (2014). Mobile medical and health apps: State of the art, concerns, regulatory control and certification. *Online Journal of Public Health Informatics*, 5(3). <https://doi.org/10.5210/ojphi.v5i3.4814>
- Hu, F., Xiao, Y., & Hao, Q. (2009). Congestion-aware, loss-resilient bio-monitoring sensor networking for mobile health applications. *IEEE Journal on Selected Areas in Communications*, 27(4), 450–465. <https://doi.org/10.1109/jsac.2009.090509>
- Sama, P. R., Eapen, Z. J., Weinfurt, K. P., Shah, B. R., & Schulman, K. A. (2014). An evaluation of mobile health application tools. *JMIR mHealth and uHealth*, 2(2), e19. <https://doi.org/10.2196/mhealth.3088>
- Siedner, M. J., Lankowski, A., Musinga, D., Jackson, J., Muzoora, C., Hunt, P. W., Martin, J. N., Bangsberg, D. R., & Haberer, J. E. (2012). Optimizing network connectivity for mobile health technologies in sub-Saharan Africa. *PLoS ONE*, 7(9), e45643. <https://doi.org/10.1371/journal.pone.0045643>
- Manshaei, M. H., Zhu, Q., Alpcan, T., Başçar, T., & Hubaux, J.-P. (2013). Game theory meets network security and privacy. *ACM Computing Surveys*, 45(3), 1–39. <https://doi.org/10.1145/2480741.2480742>
- McNab, C. (n.d.). Know your network: Network security assessment. <https://ftp.tex-host.ru/Security/%D0%94%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D1%84%D0%B0%D0%B9%D0%BB%D1%8B/Reading/08%20IDS/IP%20Network%20Scanning.pdf>
- Shyam Nandan Kumar. (2015). Review on network security and cryptography. *International Transaction of Electrical and Computer Engineers System*, 3(1), 1–11. <https://pubs.sciepub.com/iteces/3/1/1/index.html>
- Karygiannis, T., & Owens, L. (n.d.). Wireless network security: Special publication 800-48. https://all.net/books/standards/NIST-CSRC/csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf

Network Security: History, Importance, and Future. (n.d.).
<https://askcypert.org/sites/default/files/Network%20Security.pdf>

Psounis, K. (1999). Active networks: Applications, security, safety, and architectures. *IEEE Communications Surveys*, 2(1). <https://bpb-usw1.wpmucdn.com/sites.usc.edu/dist/b/364/files/2019/05/actnet.pdf>

Khan, S. A. (2022). Modern network security: Issues and challenges.
https://www.academia.edu/74861404/MODERN_NETWORK_SECURITY_ISSUES_AND_CHALLENGES

Marin, G. A. (2005). Network security basics. *IEEE Security and Privacy Magazine*, 3(6), 68–72.
<https://doi.org/10.1109/msp.2005.153>

Marias, G. F., Barros, J., Fiedler, M., Fischer, A., Hauff, H., Herkenhoener, R., Grillo, A., Lentini, A., Lima, L., Lorentzen, C., Mazurczyk, W., Meer, H., Oliveira, P. F., Polyzos, G. C., Pujol, E., Szczypiorski, K., Vilela, J. P., & Vinhoza, T. T. V. (2011). Security and privacy issues for the network of the future. *Security and Communication Networks*, 5(9), 987–1005. <https://doi.org/10.1002/sec.384>

Saxena, N., Grijalva, S., Chukwuka, V., & Vasilakos, A. V. (2017). Network security and privacy challenges in smart vehicle-to-grid. *IEEE Wireless Communications*, 24(4), 88–98.
<https://doi.org/10.1109/mwc.2016.1600039wc>

Wang, H., Wang, Y., Taleb, T., & Jiang, X. (2019). Editorial: Special issue on security and privacy in network computing. *World Wide Web*. <https://doi.org/10.1007/s11280-019-00704-x>

Zhang, C., Sun, J., Zhu, X., & Fang, Y. (2010). Privacy and security for online social networks: Challenges and opportunities. *IEEE Network*, 24(4), 13–18. <https://doi.org/10.1109/mnet.2010.5510913>

Winkler, T., & Rinner, B. (2014). Security and privacy protection in visual sensor networks. *ACM Computing Surveys*, 47(1), 1–42. <https://doi.org/10.1145/2545883>

Gao, Y. (2018). Mobile network security and privacy in WSN. *Procedia Computer Science*, 131, 647–653. <https://www.sciencedirect.com/science/article/pii/S187705091830320X/pdf>

Chan, H., & Perrig, A. (2003). Security and privacy in sensor networks. *Computer*, 36(10), 103–105.
<https://doi.org/10.1109/MC.2003.1236475>

Jain, A. K., Sahoo, S. R., & Kaubiya, J. (2021). Online social networks security and privacy: Comprehensive review and analysis. *Complex & Intelligent Systems*, 7(5), 2157–2177.
<https://doi.org/10.1007/s40747-021-00409-7>

Zeng, K., Govindan, K., & Mohapatra, P. (2010). Non-cryptographic authentication and identification in wireless networks. *IEEE Wireless Communications*, 17(5), 56–62.
<https://doi.org/10.1109/mwc.2010.5601959>

Shah, H., Kakkad, V., Patel, R., & Doshi, N. (2019). A survey on game theoretic approaches for privacy preservation in data mining and network security. *Procedia Computer Science*, 155, 686–691. <https://www.sciencedirect.com/science/article/pii/S1877050919310142/pdf>

Khelifi, H., Luo, S., Nour, B., & Shah, S. C. (2018). Security and privacy issues in vehicular named data networks: An overview. *Mobile Information Systems*, 2018, 1–11. <https://doi.org/10.1155/2018/5672154>

Toch, E., Bettini, C., Shmueli, E., Radaelli, L., Lanzi, A., Riboni, D., & Lepri, B. (2018). The privacy implications of cyber security systems. *ACM Computing Surveys*, 51(2), 1–27. <https://doi.org/10.1145/3172869>

Bambauer, D. (2013). Privacy versus security. <https://scholarlycommons.law.northwestern.edu/cgi/viewcontent.cgi?article=7454&context=jclc>

Beach, A., Gartrell, M., & Han, R. (n.d.). Solutions to security and privacy issues in mobile social networking. https://home.cs.colorado.edu/~rhan/Papers/SMW09_Secure_Mosonets.pdf

IEEE Xplore Full-Text PDF. (2026). <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5755600>
Alrasd, A. (2008). Network security: Comparison of WEP, WPA, and RSN security protocols. <https://www.academia.edu/2470176/>

Malik, M. S. A., Ahmed, M., Abdullah, T., Kousar, N., Nigar, M., & Awais, M. (2018). Wireless body area network security and privacy issue in e-healthcare. *International Journal of Advanced Computer Science and Applications*, 9(4). <https://doi.org/10.14569/ijacsa.2018.090433>

Chen, X., Makki, K., Yen, K., & Pissinou, N. (2009). Sensor network security: A survey. *IEEE Communications Surveys & Tutorials*, 11(2), 52–73. <https://doi.org/10.1109/surv.2009.090205>

Li, M., Lou, W., & Ren, K. (2010). Data security and privacy in wireless body area networks. *IEEE Wireless Communications*, 17(1), 51–58. Fasbender, A., Kesdogan, D., & Kubitz, O. (1996). Analysis of security and privacy in mobile IP. <https://www.scss.tcd.ie/hitesh.tewari/papers/fasbender.pdf>

Abdulhamid, S. M., Ahmad, S., Waziri, V. O., & Jibril, F. N. (2011). Privacy and national security issues in social networks. <https://arxiv.org/abs/1402.3301>

Li, J. (2018). Cyber security meets artificial intelligence: A survey. *Frontiers of Information Technology & Electronic Engineering*, 19(12), 1462–1474. <https://doi.org/10.1631/fitee.1800573>

Mattos, D. M. F., Krief, F., & Rueda, S. J. (2020). Blockchain and artificial intelligence for network security. *Annals of Telecommunications*, 75(3–4), 101–102. <https://doi.org/10.1007/s12243-020-00754-7>

Perez Veiga, A. (2018). <https://arxiv.org/pdf/1803.09992>

Das, R., & Sandhane, R. (2021). Artificial intelligence in cyber security. *Journal of Physics: Conference Series*, 1964(4). <https://doi.org/10.1088/1742-6596/1964/4/042072>

- Abbas, N. N., Ahmed, T., Shah, S. H. U., Omar, M., & Park, H. W. (2019). Investigating the applications of artificial intelligence in cyber security. *Scientometrics*, 121(2), 1189–1211. <https://doi.org/10.1007/s11192-019-03222-9>
- M. Sumathi, Shruthi, J., Jain, V., Kumar, G. K., & Khan, Z. Z. (2023). Using artificial intelligence and IoT for improving network security by hybrid cryptography approach. *Evergreen*, 10(2), 1133–1139. <https://doi.org/10.5109/6793674>
- Zaman, S., Alhazmi, K., Aseeri, M. A., Ahmed, M. R., Khan, R. T., Kaiser, M. S., & Mahmud, M. (2021). Security threats and artificial intelligence-based countermeasures for IoT networks: A comprehensive survey. *IEEE Access*, 9, 94668–94690. <https://doi.org/10.1109/ACCESS.2021.3089681>
- Wu, H., Han, H., Wang, X., & Sun, S. (2020). Research on artificial intelligence enhancing Internet of Things security: A survey. *IEEE Access*, 8, 153826–153848. <https://doi.org/10.1109/access.2020.3018170>
- IEEE Xplore Full-Text PDF. (2024). <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9797706>
- IEEE Xplore Full-Text PDF. (2024). <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10403908>
- Logeshwaran, J. (n.d.). AICSA – An artificial intelligence cyber security algorithm for cooperative P2P file sharing in social networks. https://ictactjournals.in/paper/IJDSML_Vol_3_Issue_1_Paper_2_251_253.pdf
- Zhang, Z., Hamadi, H. A., Damiani, E., Yeun, C. Y., & Taher, F. (2022). Explainable artificial intelligence applications in cyber security: State-of-the-art in research. *IEEE Access*, 10, 93104–93139. <https://doi.org/10.1109/access.2022.3204051>
- Peyo. (2010). *Les p'tits Schtroumpfs*. Hachette Jeunesse. Haider, N., Baig, M. Z., & Imran, M. (2020). Artificial intelligence and machine learning in 5G network security: Opportunities, advantages, and future research trends. <https://doi.org/10.48550/arXiv.2007.04490>
- Horowitz, T. (2018). The high prevalence effect meets the low prevalence effect. *Journal of Vision*, 18(10), 519. <https://doi.org/10.1167/18.10.519>
- Making sure you're not a bot! (2026). <https://hal.science/hal-01959251/document>
- Attkan, A., & Ranga, V. (2022). Cyber-physical security for IoT networks. *Complex & Intelligent Systems*, 8, 3559–3591. <https://doi.org/10.1007/s40747-022-00667-z>
- Boukraa, L., Essahraoui, S., El Makkaoui, K., Ouahbi, I., & Esbai, R. (2023). Intelligent intrusion detection in software-defined networking. *Procedia Computer Science*, 224, 26–33. <https://doi.org/10.1016/j.procs.2023.09.005>
- IEEE Xplore Full-Text PDF. (2026). <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8823861>

Li, S. (2021). Application analysis of artificial intelligence in library network security. *Journal of Physics: Conference Series*, 1744(3), 032024. <https://doi.org/10.1088/1742-6596/1744/3/032024>

Omrani, T., Dallali, A., Chibani Rhaimi, B., & Fattahi, J. (2018). Fusion of ANN and SVM classifiers for network attack detection. <https://doi.org/10.48550/arXiv.1801.02746>

Hu, Y., Kuang, W., Qin, Z., Li, K., Zhang, J., Gao, Y., Li, W., & Li, K. (2021). Artificial intelligence security: Threats and countermeasures. *ACM Computing Surveys*, 55(1), 1–36. <https://doi.org/10.1145/3487890>

IEEE Xplore Full-Text PDF. (2026). <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9878067>

Saber, M., El Rharras, A., Saadane, R., Aroussi, H. K., & Wahbi, M. (2022). Artificial neural networks, support vector machine and energy detection for spectrum sensing. *International Journal of Communication Networks and Information Security*, 11(1). <https://doi.org/10.17762/ijcnis.v11i1.3718>

Matlou, O. G., & Abu-Mahfouz, A. M. (2017). Utilising artificial intelligence in software-defined wireless sensor networks. *IECON 2017*, 6131–6136. <https://doi.org/10.1109/IECON.2017.8217065>

Dalal, S., Lilhore, U. K., Foujdar, N., Simaiya, S., Ayadi, M., Almujaally, N., & Ksibi, A. (2023). Next-generation cyber attack prediction for IoT systems. *Journal of Cloud Computing*, 12(1). <https://doi.org/10.1186/s13677-023-00517-4>