

RESOURCE OPTIMIZATION IN HETEROGENEOUS
DISTRIBUTED DATA STREAM MINING WITH
PERFORMANCE ASSESSMENT ON ASTHMA
HOSPITALIZATION PREDICTIVE MODELING

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Supervisors

Dr Farhaan Mirza

Dr M. Asif Naeem

Dr Amy Hai Yan Chan (Clinical Advisor)

October 2022

By

Rashi Bhalla

Data Science Research Centre

School of Engineering, Computer and Mathematical Sciences

Abstract

The Big Data Era has presented many opportunities for data mining techniques to discover knowledge patterns across an exponentially growing diverse collection of data. In many application domains, data exists in a distributed fashion across geographical locations; as such, the nature of the data collected by each location may differ from its peer nodes in its network, thus causing heterogeneity in the data. Such scenarios of distributed databases require mining methods that are distinct from traditional homogeneous distributed databases where the structure is identical across locations. The current demand is to analyze real-time data; consequently, stream computing is becoming a popular choice. Data generated continuously at a high pace at distributed sites or locations are termed as distributed data streams. Recent approaches toward Distributed Data Mining (DDM) have focused on addressing the heterogeneous nature of data sources. However, such approaches do not prioritize the reduction of data communication costs which could be prohibitive in large-scale sensor networks where bandwidth is a limited resource. In fact, higher communication and computational costs are the two most prominent problems encountered in heterogeneous distributed environments. An effort to decrease communication in the distributed environment adversely influences classification accuracy; therefore, a research challenge lies in maintaining a balance between transmission cost, computational cost, and accuracy.

This research covers the heterogeneous distributed data mining problem, extendable to the case where data arrives continuously in streaming mode. We propose a suite of

algorithms to address specific issues in mining data from heterogeneous distributed streaming settings. Our experimental testing reveals that performance efficiency can be achieved across a wide range of datasets. The first algorithm, *Performance Optimizer in Distributed Stream Mining* (PODSM), having its roots in Bayesian Inference, is targeted towards reducing the communication volume and resource time in a heterogeneous DDM environment while retaining prediction accuracy. A reduction of 34.66% in communication was obtained for one of the datasets with nearly 27% savings in resource time. The second algorithm, *Minimized Tree for Distributed Mining* (MTDM), presents an efficient and robust method for learning the relationship between various distributed sites using a tree. In this regard, a saving of 37.65% in resource time has been reported for one dataset while improving the accuracy by 1.33%. To assess the algorithms' competency, we validated them on a case study built using real datasets from real-world sources to predict demands for asthma-related emergency hospitalizations into Low or High classes. Considerable savings in terms of communication and resource time were attained upon execution of PODSM and MTDM while preserving accuracy levels, thus portraying their potential to achieve a good trade-off between accuracy and resource utilization. The study concludes that PODSM and MTDM are proficient in conjoint servicing heterogeneous distributed data sources in any resource-constrained scenario. Moreover, the capability of the algorithms to maintain a balance between accuracy, communication, and resource time makes them flexible enough for a diverse range of applications.

Contents

Abstract	2
Attestation of Authorship	12
Publications	13
Acknowledgements	14
1 Introduction	17
1.1 Overview	18
1.2 Background	23
1.3 Motivation	28
1.4 Problem Statement	32
1.5 Research Questions	34
1.6 Research Objectives	37
1.7 Thesis Contributions	39
1.8 Outline of the Thesis	41
2 Literature Review	44
2.1 Introduction	45
2.2 Distributed Data Mining in Offline Context	47
2.2.1 Approaches for homogeneously distributed data	51
2.2.2 Approaches for heterogeneously distributed data	53
2.2.3 Approaches applicable to both distribution of data	54
2.2.4 Challenges in DDM	56
2.3 Data Stream Mining	58
2.3.1 Processing data streams	59
2.3.2 Classification	63
2.3.3 Ensemble learning in data streams	69
2.3.4 Challenges in data stream mining	71
2.4 Heterogeneous Distributed Data Stream Mining	72
2.4.1 Sampling in heterogeneous data streams	72
2.4.2 Integration of heterogeneous data sources	73
2.4.3 Approaches for mining heterogeneously distributed streams	74

2.4.4	Challenges in HDDSM	80
2.5	Predictive Modeling via Heterogeneous Sources for Asthma	81
2.5.1	Individual based models	82
2.5.2	Population based models	94
2.6	Background Work	97
2.7	Research Gaps	101
2.8	Chapter Summary	105
3	Performance Optimizer in Distributed Stream Mining (PODSM)	106
3.1	Introduction	107
3.2	Problem Definition	107
3.3	Architecture	110
3.3.1	PODSM algorithm	116
3.3.2	Illustration of transmission protocol	120
3.3.3	Aggregation methods	124
3.4	Experimental Evaluation	125
3.4.1	Experimental setup	125
3.4.2	Performance assessment metrics	127
3.4.3	Performance comparison	128
3.4.4	Parameter sensitivity analysis of PODSM	136
3.5	Applicability and Limitations of PODSM	138
3.6	Chapter Summary	140
4	Minimized Tree for Distributed Mining (MTDM)	141
4.1	Introduction	142
4.2	Problem Definition	142
4.3	Architecture	148
4.3.1	MTDM algorithm	152
4.3.2	Complexity analysis	156
4.3.3	Aggregation methods	160
4.4	Experimental Evaluation	161
4.4.1	Experimental setup	161
4.4.2	Performance assessment metrics	162
4.4.3	Performance comparison	164
4.4.4	Parameter sensitivity analysis of MTDM	171
4.5	Applicability and Limitations of MTDM	175
4.6	Chapter Summary	175
5	Case Study: Predicting Peak Demand Days for Asthma-Related ED Visits and Hospitalizations	177
5.1	Introduction	178
5.2	Problem Definition	179
5.3	Stages in Predictive Modeling	182
5.3.1	Data collection from diverse sources	183

5.3.2	Data preparation	189
5.3.3	Feature selection	193
5.3.4	Prediction model	194
5.4	Experimental Evaluation	198
5.5	Applicability and Limitations	208
5.6	Chapter Summary	209
6	Implementation of PODSM and MTDM on the asthma case study	211
6.1	Introduction	212
6.2	Problem Definition	213
6.3	Proposed Methodology	214
6.4	Experimental Evaluation	219
6.4.1	Experiment configuration	219
6.4.2	Performance assessment metrics	220
6.4.3	Performance evaluation	220
6.5	Applicability and Limitations	239
6.6	Chapter Summary	240
7	Conclusion	241
7.1	Research Achievements	242
7.2	Variations of Work Proposed in this Study	246
7.3	Limitation of this Research Work	248
7.4	Future Work	250
	References	254
	Appendices	281

List of Tables

2.1	Difference between traditional and stream processing.	59
2.2	Some data stream classifiers with advantages and disadvantages. . . .	66
2.3	Summarization of individual-level studies for asthma.	91
2.4	Existing similar work with an aim to develop a population-based prediction model for asthma.	97
2.5	Comparison among the state-of-the-art foundational work.	100
3.1	Summarization of transmission and processing status at trouble site for different possibilities of confidence status for three sites; A, B, and C.	109
3.2	Properties of datasets used for performance evaluation.	127
3.3	Explanation of metrics used for performance assessment.	128
3.4	Accuracy comparison between state-of-the-art methods and PODSM.	129
3.5	Comparison between state-of-the-art methods and PODSM in terms of communication.	130
3.6	Performance evaluation for OCC dataset in terms of Precision and Recall.	130
3.7	Resource time comparison between HDSM and PODSM.	132
3.8	Resource time comparison for different aggregation methods.	132
3.9	Impact of change in the size of <i>batch</i> -1 on accuracy and communication.	138
3.10	Effect of changing the frequency of updating the statistics.	139
4.1	Performance evaluation dataset properties.	163
4.2	Explanation of metrics used for performance evaluation.	163
4.3	Comparison in terms of accuracy.	165
4.4	Comparison in terms of communication, combinations, and TCM.	166
4.5	Comparison between centralized mining and MTDM.	166
4.6	Resource time comparison in terms of TFTS.	167
4.7	Comparison for other aggregation methods.	167
4.8	Impact of changing values of sensitive parameters.	173
5.1	Frequency of readings.	190
5.2	Analytical summary of predictors.	202
5.3	Comparative analysis of different Machine Learning algorithms.	203
5.4	Comparison of performance achieved through implementing different algorithms over-sampled data.	205
6.1	Evaluation of state-of-the-art methods.	222

6.2	Comparison between HDSM and PODSM for different aggregation methods.	223
6.3	<i>Batch</i> -1 sensitivity analysis in terms of MTCP and MTBR.	230
6.4	Sensitivity analysis of PODSM on the asthma case study by changing the frequency of updating statistics.	231
6.5	Results after implementation of MTDM.	232
6.6	Comparison between HDSM and MTDM in terms of TC, TCM, and TFTS.	233
6.7	Impact of changing values of sensitive parameters.	237

List of Figures

1.1	Architecture for processing distributed sources of data	19
1.2	Partitioning of data into two types: homogeneous and heterogeneous .	21
1.3	Example demonstrating cross-terms	25
1.4	Diagrammatic presentation of cross-terms	26
1.5	Types of concept drifts	27
1.6	Conceptual framework	35
1.7	Research contributions	39
1.8	Research contributions	40
1.9	Thesis roadmap	43
2.1	Literature review focus areas	46
2.2	Two types of predictive modeling for asthma	83
2.3	Mapping of research gaps to research questions to research contributions	104
3.1	Decentralized architectures for processing distributed sources	110
3.2	Pictorial representation of the architecture used in PODSM, consisting of primary sites, trouble sites, and the aggregator.	112
3.3	Structural illustration of sites for calculating statistics	113
3.4	A flow diagram summarizing the role of the primary site in PODSM architecture.	121
3.5	Significance of PODSM shown through graphical representation in terms of accuracy and communication	131
3.6	Accuracy and transmission comparison between HDSM and PODSM for all aggregation methods	133
3.7	Experimental timeline for HDSM	135
3.8	Timeline visualizations for PODSM experiment	136
3.9	A ranking comparison depicting competency to maintain the trade-off between accuracy and communication between centralized experiment and PODSM configured for TF 1x, 1.5x, and 2x.	137
4.1	A single trouble site versus hierarchy of trouble sites	144
4.2	Initialization of creation monitors for a structure consisting of nine distributed sites.	147
4.3	A sample centralized tree.	149
4.4	Vertically distributed sites structure.	150

4.5	Pictorial representation of tree after parsing level 1.	151
4.6	Graphic presentation of a subtree in a tree.	154
4.7	Demonstration of the tree, the two lists, and variable <i>maximum_trouble_records</i> , upon completion of parsing Level 1 of the original tree.	155
4.8	Example depicting pre-pruning condition of MTDM	156
4.9	Pictorial representation of algorithm MTDM.	158
4.10	Complexity analysis of parsing the tree.	160
4.11	Timelines of accuracy, data transmission, and resource time with HDSM.	169
4.12	Timelines of accuracy, data transmission, and resource time after implementing MTDM.	170
4.13	Relative ranks in HDSM.	171
4.14	Relative ranking in MTDM.	172
4.15	Differentiation between centralized experiment, HDSM, and MTDM.	172
4.16	Impact of changing the value of minimum records.	174
5.1	Asthma mortality rate from 2000-2015 with age-adjusted by Asthma Respiratory Foundation NZ (2019).	180
5.2	Asthma hospitalizations per 100,000 people per year for the period of 2000-2017 by Barnard and Zhang (2019).	181
5.3	Risk assessment for asthma-related hospitalizations using diverse sources.	183
5.4	Processing heterogeneous data sources for generating the forecasting model.	184
5.5	Geographic sites for gathering meteorological data.	187
5.6	Representation of sites on the map used for collecting air quality data.	188
5.7	Graphs depicting the data values for Google search trends filtered using keywords related to asthma from a period of 1st Jan 2018 to 1st Jan 2021.	189
5.8	Plots for presenting outliers	191
5.9	Comparison of raw and processed data	192
5.10	Lagged effect of UVI for a period of 0 to 4 days.	194
5.11	Lagged effect calculation for all parameters	195
5.12	Ensemble modeling using three base models with final classification generated using majority voting mechanism.	197
5.13	a) Plot for daily counts of asthma-related ED visits and admissions b) Monthly visualizations of the hospital data from 01/2018 to 12/2020 c) Timeline for actual and predicted values for monthly rate of ED visits and admissions using ARIMA for a test period from 07/2019 to 12/2020.	200
5.14	Correlation calculation	201
5.15	Feature importance by XGB classifiers.	203
5.16	Accuracy curves	204
5.17	Sampling techniques	206
5.18	Accuracy curves for balanced dataset	207
5.19	Learning curves for training and cross-validation phases.	208

6.1	Pictorial representation stating the methodologies that would be applied to the asthma case study.	216
6.2	Pictorial representation stating the methodologies that would be applied to the asthma case study.	218
6.3	Timeline for HDSM	224
6.4	Timeline for PODSM.	225
6.5	PODSM's impact on communication for different sizes of the dataset.	226
6.6	Accuracy and communication evaluation when the size of <i>batch-1</i> is 10000.	228
6.7	Effect of <i>batch-1</i> size attaining a value of 17000 on accuracy and communication.	229
6.8	Accuracy and communication ranking between centralized and PODSM.	232
6.9	Timeline for HDSM for comparing it with MTDM.	234
6.10	Timeline for MTDM.	235
6.11	Resource time comparison between HDSM and MTDM	236
6.12	Footprint of the number of records on TC and resource time for processing a tree.	238

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 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.

A handwritten signature in black ink that reads "Rashi Bhalla". The script is cursive and fluid, with the first letters of each name being capitalized and prominent.

Signature of candidate

Publications

- Bhalla, R. & Pears, R. & Naeem, M. A. & Mirza, F. (2022). Novel method for optimizing performance in resource constrained distributed data streams. *Applied Intelligence*. <https://doi.org/10.1007/s10489-021-03019-5>
- Bhalla, R. & Pears, R. & Naeem, M. A. & Mirza, F. (2022). Scalable and Efficient Method for Learning the Relationship between Heterogeneous Distributed Data Streams Using a Minimized Tree. Manuscript in review for publication to *Data Mining and Knowledge Discovery*.
- Bhalla, R. & Mirza, F. & Naeem, M. A. & Chan, A. H. Y. (2022). Predicting Peak Demand Days for Asthma-Related Emergency Hospitalisations. Manuscript in review for publication to *BMC Medical Informatics and Decision Making*.
- Bhalla, R. & Mirza, F. & Naeem, M. A. (2022). Online Processing of Heterogeneous Data Sources for Predicting Peak Demand Days for Asthma-Related Emergency Hospitalizations. Manuscript in review for publication to *Machine Learning*.

Acknowledgements

Words cannot express my gratitude to the Almighty God for bestowing his blessings on me and providing me with the priceless treasure of life through my course of PhD. I could not have undertaken my PhD without the support of my supervisory team. I'm so grateful to my primary supervisor, Dr Farhaan Mirza, for his consistent support and feedback that has helped me to accomplish my research goals and kept me moving further. He has been a great source of encouragement and motivation. My sincere thanks would also go to my secondary supervisor, Dr Muhammad Asif Naeem, for his kind support and insightful suggestions. I really appreciate their time and concerns, and I would also like to acknowledge my former primary supervisor, Dr Russel Pears, for his extensive knowledge and guidance in learning about data mining.

I would like to express my deepest appreciation to Dr Amy Hai Yan Chan from the University of Auckland and Auckland District Health Board for helping me get real data on asthma-related ED visits and hospitalizations which was used as a case study in my research work. I am also thankful for her input in clinical expertise. I'd like to acknowledge Louis Boamponsem from Auckland Council and NIWA for providing data on Air Quality and meteorological parameters. I would like to extend my sincere thanks to Auckland Health Research Ethics Committee (AHREC) for approving my ethics application entitled "A Practical Tool to Predict Asthma-Related Emergency Department Visits and Admissions at Population Level" on 23/04/2021.

I could not have embarked upon this journey without the support of Auckland University of Technology and the School of Engineering, Mathematical, and Computer Sciences. Your financial support, excellent facilities, and immense support from colleagues, including administrative help from Karishma Bhat and members of the Graduate Research Group and technical support from ICT, have made this thesis all the richer. Thanks should also go to my fellow students and friends, with a special appreciation to Ben Denham.

This endeavor would not have been possible without the love, support, encouragement, and prayers of my family. I'm extremely grateful to my husband, Sumit, who has accompanied me in all ups and downs of my progress in PhD. My daughter, Radhya, for being patient, cooperative, and maturely understanding the requirement of time. Without you, I would not be where I am today. I am deeply indebted to my parents for the sacrifices they made for me to get an education and for always teaching me the value of education. My late father's confidence in me has always inspired me to never give up and work hard in life. My mother's warmth and prayers have helped me to reach this milestone.

Acronyms

Acronym	Definition
ADHB	Auckland District Health Board
AI	Artificial Intelligence
ANN	Artificial Neural Networks
ARF	Adaptive Random Forest
ARIMA	Autoregressive Integrated Moving Average
COPD	Chronic Obstructive Pulmonary Disease
CPM	Clinical Prediction Models
DDM	Distributed Data Mining
ED	Emergency Department
EHR	Electronic Health Record
HDDSM	Heterogeneous Distributed Data Stream Mining
HDSM	Hierarchical Distributed Stream Miner
IoMT	Internet of Medical Things
IoT	Internet of Things
MDSM	Mobile Data Stream Mining
MLP	Multi-Layer Perceptron
MTBR	Mean Time Between Completions
MTCP	Mean Time on Critical Path

MTDM	Minimized Tree for Distributed Mining
NZ	New Zealand
OECD	Organization for Economic Co-operation and Development
PODSM	Performance Optimizer in Distributed Stream Mining
RF	Random Forest
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machines
TC	Total Combinations
TCM	Time for Creation Monitors
TF	Trouble Factor
TFTS	Time for First Trouble Site
TT	Total Transmission
TUCM	Time for Updating Creation Monitors
UVI	UltraViolet Index
WSN	Wireless Sensor Networks
XGB	Extreme Gradient Boosting

Chapter 1

Introduction

This introductory chapter raises the curtain of the subject matter by introducing the reader to the foci. It forms the preparatory session that provides the rationale behind this study. The chapter comprises various sections that establish the territory by highlighting the topic's relevance. The later sections help identify the niche by indicating the research problem, further listing the purpose and motivation of this research in occupying the niche. The research contributions made through this work have been comprehensively presented using illustrations.

1.1 Overview

The volume of data in the world is exploding in both scientific and commercial domains (Fuqiang, 2011; Ilin & Klimin, 2019). As the size of raw data is increasing, parallelly data mining algorithms are becoming a necessity. The *data mining process* aims at extracting valuable information from large datasets and has its foundation in databases, statistics, machine learning, and data visualization (Geng & Hamilton, 2006; Marban, Mariscal & Segovia, 2009; Borgi, Zoghلامي, Abed & Mohamed Saber, 2017). Recommendation systems that lie at the heart of almost every modern commercial application utilize data mining tools to predict consumer behavior. These systems are mainly used to enhance sales or improve user experience by making suggestions proactively according to user choices; for instance, they are used by Netflix to predict users' movie choices. Data mining includes various algorithms and methods that can transform an enormous mountain of data into nuggets of knowledge, including *supervised*, *unsupervised*, and *semi-supervised learning* techniques (IBM Cloud Education, 2021).

Data is constantly emerging and is often distributed in nature (Garofalakis, 2009; Chmaj & Selvaraj, 2015; Mikalef, Boura, Lekakos & Krogstie, 2019). In many application domains, data is distributed across different geographical locations (Devi, 2014; Boutaba et al., 2018). The classic *data warehouse-based architecture* works by uploading all the data into the warehouse for centralized processing, as shown in Figure 1.1a. However, as the data sizes grew and the applications became more complex, there is a need to replace centralized data mining with parallel and distributed techniques. Moreover, due to the limitations in classic data warehouse-based architecture, such as extended response time, lack of proper use of distributed resources, high communication cost, and privacy issues, this architecture could not achieve effective outcomes in distributed environments and therefore gave rise to *distributed processing architecture* (da Silva, Giannella, Bhargava, Kargupta & Klusch, 2005; Skillicorn & McConnell,

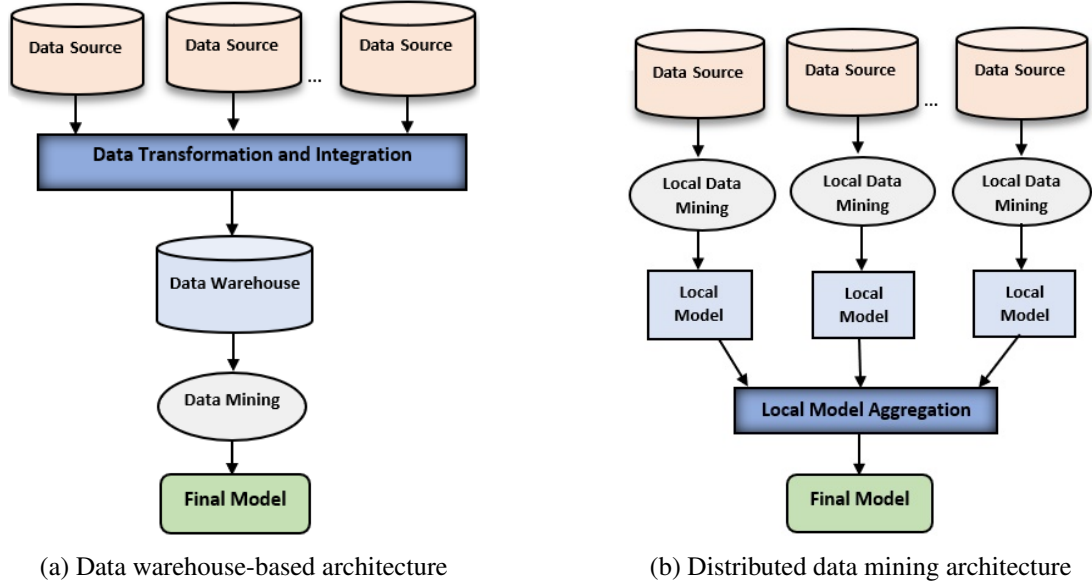


Figure 1.1: Frameworks for processing distributed sources of data; in a) form of processing data from diverse sources is agglomerated at a fusion site before applying data mining methods, whereas in b) architecture data mining methods are applied on each local site, thus generating local models which are then combined to obtain a final model.

2008; Gan, Lin, Chao & Zhan, 2017). A classic warehouse-based architecture, when implemented in distributed scenarios, requires the transmission of all the data from all the sites to a single centralized site, which involves huge communication costs and necessitates processing at a single site. Furthermore, in this architecture, privacy concerns are raised as the data values flow between the sites instead of the data mining results.

Figure 1.1b presents a Distributed Data Mining (DDM) framework where each site processes locally available data, and finally, local models are agglomerated to form a global model (Kargupta, Byung-Hoon, Hersherberger & Johnson, 1999; Weinberg & Last, 2017). The distributed approach is more effective in terms of computation, storage, privacy, and cost of communication (Gorawski & Pluciennik-Psota, 2009; Zeng et al., 2012; Jethva, 2022). An example that can be used here to draw the difference between both architectures would be the New Zealand (NZ) healthcare system. A person identified by a unique key (such as National Health Index or commonly known

as NHI number in NZ) may be using health services from primary, secondary, or tertiary sectors. Such that each of these services acts as a separate site containing different sets of attributes/features but sharing a common key attribute which is NHI in this case. If data from all these services are accumulated at a central site for generating a piece of information about a person by joining record fragments from different sites using the common key attribute, then the architecture would be classified as data warehouse-based. On the other hand, if all these sites or services use their local mining algorithms to generate their local models, which are aggregated instead of aggregating data to come up with a combined final model, would be a DDM architecture.

Data can be distributed in one of two ways, either they follow *homogeneous (or horizontal) partitioning* or *heterogeneous (or vertical) partitioning* (Park & Kargupta, 2003). *Homogeneous partitioning* (Tennant, Stahl, Rana & Gomes, 2017; Weinberg & Last, 2017) is the most common way of collecting data, according to which the features (also known as attributes) describing the data are the same across each distributed site and records get distributed among sites. On the other hand, under *heterogeneous partitioning* (Akbar et al., 2018), the attributes present at distributed sites differ from each other for the same set of records. *For example*, a toy retail outlet collecting data about its different products can be distributed across sites following either homogeneous or heterogeneous partitioning. Figure 1.2a renders the presentation when data gets horizontally distributed across different products, whereas heterogeneous or vertical distributions shown in Figure 1.2b split the data using the product features.

In this “*Information Explosion Era*”, the input to the system is an unbounded stream of data instances arriving in real-time, and in some cases, at high speed (De Francisci Morales, Bifet, Khan, Gama & Fan, 2016). Credit card transactions, Google searches, and a plethora of others covering business, scientific, and security applications (L. Zhang, Xie, Xidao & Zhang, 2018) are some fields that necessitate data analysis in real-time or near real-time as they arrive. Data stream mining algorithms that have

Key	Product Name	Description	Stocks	Price
AA001	Lego -500	Farm Lego -500 Pieces	20	15
AA004	Bowling Game	Bowling pins -6	15	20
AA005	Football	Football – Brown, big	5	30
AA120	Puzzle	Puzzle- Animals 200 Pieces	10	10

Key	Product Name	Description	Stocks	Price
AA001	Lego -500	Farm Lego -500 Pieces	20	15
AA004	Bowling Game	Bowling pins -6	15	20

Key	Product Name	Description	Stocks	Price
AA005	Football	Football – Brown, big	5	30
AA120	Puzzle	Puzzle- Animals 200 Pieces	10	10

(a) Homogeneous or horizontal partitioning

Key	Product Name	Description	Price	Stocks
AA001	Lego -500	Farm Lego - 500 Pieces	15	20
AA004	Bowling Game	Bowling pins - 6	20	15
AA005	Football	Football – Brown, big	30	5
AA120	Puzzle	Puzzle- Animals 200 Pieces	10	10

Key	Product Name	Description	Price
AA001	Lego -500	Farm Lego -500 Pieces	15
AA004	Bowling Game	Bowling pins -6	20
AA005	Football	Football – Brown, big	30
AA120	Puzzle	Puzzle- Animals 200 Pieces	10

Key	Stocks
AA001	20
AA004	15
AA005	5
AA120	10

(b) Heterogeneous or vertical partitioning

Figure 1.2: Illustration of different methods of distributing data among sites, where a) shows a partition under which records get distributed while b) shows the column-wise state of the partition.

the ability to learn from continuous streams of data records have been developed; they have the ability to scan and process each record only once and, in doing so, avoid the storage of an extensive collection of records for training, which is impractical in some cases (Bhaduri et al., 2007). In *data stream environment*, each stream is a sequence of samples such that $S = (s_1, s_2, \dots, s_n)$ where samples are seen in order, and n can be possibly infinite. In order to tackle another challenge of data stream mining, expressly

rapid change in data distribution of the input data, these algorithms are designed with the ability to adapt to changes in the underlying patterns of the data streams, also known as “*concept drift*” (Wares, Isaacs & Elyan, 2019).

Although replacing a data warehouse-based architecture with a DDM approach provides the solution to the limitations witnessed in the former. But on the other hand, a DDM procedure also poses many challenges, specifically in a vertically distributed scenario. The underlying challenge lies in maintaining a trade-off between *accuracy and resources utilized* in the mining process. As mentioned earlier, a DDM architecture prevents data accumulation at a central site as opposed to a warehouse or centralized architecture. It uses local data models (i.e., results from each local model as shown in Figure 1.1b) to generate the final output, which affects the output’s accuracy. However, a DDM architecture in a vertically distributed scenario struggles to attain an accuracy level comparable to a centralized architecture with minimum utilization of resources. Also, each site is a separate entity with some indirect relationship between them. Uncovering these relationships without data centralization is another troublesome task. These problems intensify in a streaming environment where the data originate from streaming sources.

In this thesis, we focus on *introducing a set of algorithms that can provide solutions to the above-discussed challenges while mining data from vertically distributed data streams*. Furthermore, we work towards applying these solutions in a real case study. The case study we used here includes data streams from distributed heterogeneous sources¹ related to asthma patients and exacerbation². The distributed data streams include *i) meteorological factors ii) air quality parameters iii) Google search trends*. This area is important as NZ is one of the countries with the highest prevalence and mortality rate due to asthma (Asthma Respiratory Foundation NZ, 2019). Overcrowding

¹Heterogeneous sources have been used to correspond to data streams originating from diverse IoT or other sensor networks, representing true vertically distributed data mining scenario.

²These refer to situations when asthma worsens or new symptoms occur.

of the Emergency Department (ED), especially in the cold season, is an existing problem in NZ, which brings complicated situations for asthma patients and health providers. As a result, in addition to making computational contributions in the form of proposed novel algorithms for mining data from vertically distributed data streams, we can also produce a real-world impact by applying these algorithms in a real-world asthma case study. This case study aims to *predict peak demand days for asthma-related hospitalizations*, expressly; it can classify days into either high or low asthma-related hospitalizations. This can help healthcare providers to pre-prepare for days when demands for asthma-related hospitalizations have been predicted to be high and combat adverse events like overcrowding.

1.2 Background

The vertically partitioned dataset is still rare but is becoming more prominent (Skillicorn & McConnell, 2008). It is generally believed that mining heterogeneous distributed data is more challenging and complex (Gan et al., 2017; Akbar et al., 2018). Let us use an example to demonstrate the complexity involved in mining data from heterogeneously distributed sources. A medical insurance company, *say XYZ*, aims to determine whether a particular client is a high-risk patient before offering insurance against medical expenses. For this prediction, the XYZ company uses data from two heterogeneous sources; the *medical site* (collecting medical data of patients) and the *financial site* (containing patients' financial information). Sample data for both sites have been shown in Figure 1.3 such that both sites contain different attributes (or feature sets) about the same patients. These sites share the common key attribute and class attribute (represented as *key* and *Hi Risk* in the tables) and are thus referred to as vertical partitions. The *key* attribute is used for identifying a patient, and the *class attribute* presents the classification result. If these individual sites build their local models on

their local data (data present at the site) using a decision tree algorithm, these models will reflect only their view of data as shown in Figures 1.4a and 1.4b. The local sites will generate the classification result by applying their local model, independent of each other. Even though these models do not involve any data transmission for producing the classifications, but on the other hand, they cannot achieve the objective of accurate prediction as the local models for these sites cannot capture an interaction that exists between features present at both the sites that are between, *Average Monthly Visits* and *Average Monthly Expenses* (referred to as *cross-terms*). *Cross-terms* are the patterns that can only be detected by combining views of two or more vertical partitions (Tsoumakas & Vlahavas, 2009; Denham, 2019). If the data from both the sites are centralized at a location generating a centralized model using all the data, this model would be able to capture the cross-terms between the sites represented in Figure 1.4c. The predictions from this model would be based upon the combined view of the sites and would be more accurate, but generating this centralized model would involve huge costs due to the transmission of data from both sites. Therefore, the main challenge while mining heterogeneous data streams lies in *learning the cross-terms while keeping the resources utilized to a minimum*.

The advancements in information and communication technology have spurred the appearance of distributed computing environments, thus launching data stream mining in the distributed arena (Tennant et al., 2017). In recent years, many applications have manifested the relevance of large-scale stream processing like online stock, Internet traffic, electric power grids, and scientific & engineering experiments. Algorithms that aid in DDM and data stream mining are valuable in many areas of science, engineering, and information management and are being deployed in applications like astrophysics, sensor networks, weather forecasting, and many others (Khan & Fan, 2012; ur Rehman, Liew, Wah & Khan, 2017; Akbar et al., 2018). A robust data stream mining approach should possess the capability to address the key aspects of Volume, Velocity, and



Key	Avg Monthly Visits	Medical History	Hi Risk
1	0.8	bad	Yes
2	0.9	bad	Yes
3	1.0	good	No
4	0.8	good	No
5	0.7	good	No
6	0.6	bad	Yes
7	1.4	good	No
8	1.1	good	Yes
9	1.5	good	No
10	2.1	good	Yes
11	1.0	good	No
12	1.6	good	Yes
13	1.3	good	Yes
14	0.8	good	No
15	1.4	bad	Yes
16	1.0	good	No
17	1.2	good	Yes
18	1.1	good	No
19	1.1	good	No

Medical Site

Key	Avg Med Expenses (Monthly)	Income	Hi Risk
1	53.00	39600	Yes
2	86.75	45500	Yes
3	23.25	60100	No
4	78.50	55000	No
5	55.00	31300	No
6	65.50	37400	Yes
7	28.25	44200	No
8	33.75	28100	Yes
9	14.25	25900	No
10	39.75	38200	Yes
11	27.00	35500	No
12	31.25	41600	Yes
13	29.00	57400	Yes
14	46.00	47100	No
15	27.50	42300	Yes
16	28.25	33000	No
17	33.50	36800	Yes
18	26.25	41300	No
19	25.50	33500	No

Financial Site

Figure 1.3: Sample data for the insurance problem under medical and financial sites and demonstration of cross-terms between both sites.

Volatility while processing the huge streams of data: *Volume, Velocity, and Volatility* (Rutkowski, Jaworski & Duda, 2020). The first two aspects demand online processing of a huge volume of data arriving at a high pace, whereas the third one focuses on the dynamic nature of the approach to adapt to changing data (C. Aggarwal, 2007). The occurrence of a drift marks an essential property of concept drift; it can be *abrupt*, *incremental*, *gradual*, or *recurring* (Gama, Žliobaitundefined, Bifet, Pechenizkiy & Bouchachia, 2014) as shown in Figure 1.5. In *abrupt/sudden drifts*, the prediction error and data distribution witnesses considerable changes in a short span. *Incremental drifts* characterizes concepts that emerge slowly with time. *Gradual drifts* does not witness an abrupt transition; instead, there is an evolution window in which the frequency of the previous concept becomes less, thereby transforming new concept instances in more prevalent. If the concepts reoccur after some time, then it is classified as *recurring drifts*.

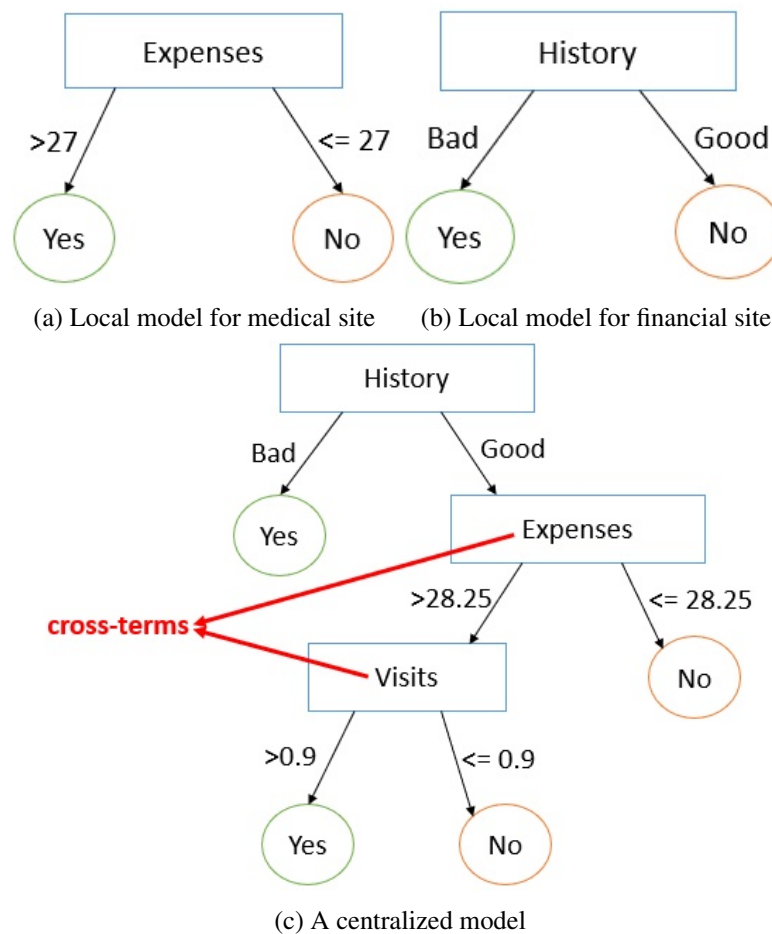


Figure 1.4: Pictorial comparison among tree models generated using local data at medical and financial sites v/s a centralized tree generated by aggregating the data from both sites to a centralized site.

While handling concept drifts, one vital point is to differentiate true drifts from outliers or noise (Chandola, Banerjee & Kumar, 2009).

Let us assume a case study of traffic monitoring to analyze the different types of concept drifts. Usually, the peak hours for traffic are during the morning and evening due to the office and school hours compared to other times in the day. The trend of high and low traffic occurs in a cycle during weekdays, which can be classified as "**recurring concept**". Due to COVID-19, lockdown restrictions were imposed in most countries leading to "**sudden or abrupt drifts**". The situations like long weekends or holidays can cause "**gradual drifts**" in traffic movements. If the number of COVID-19 cases is

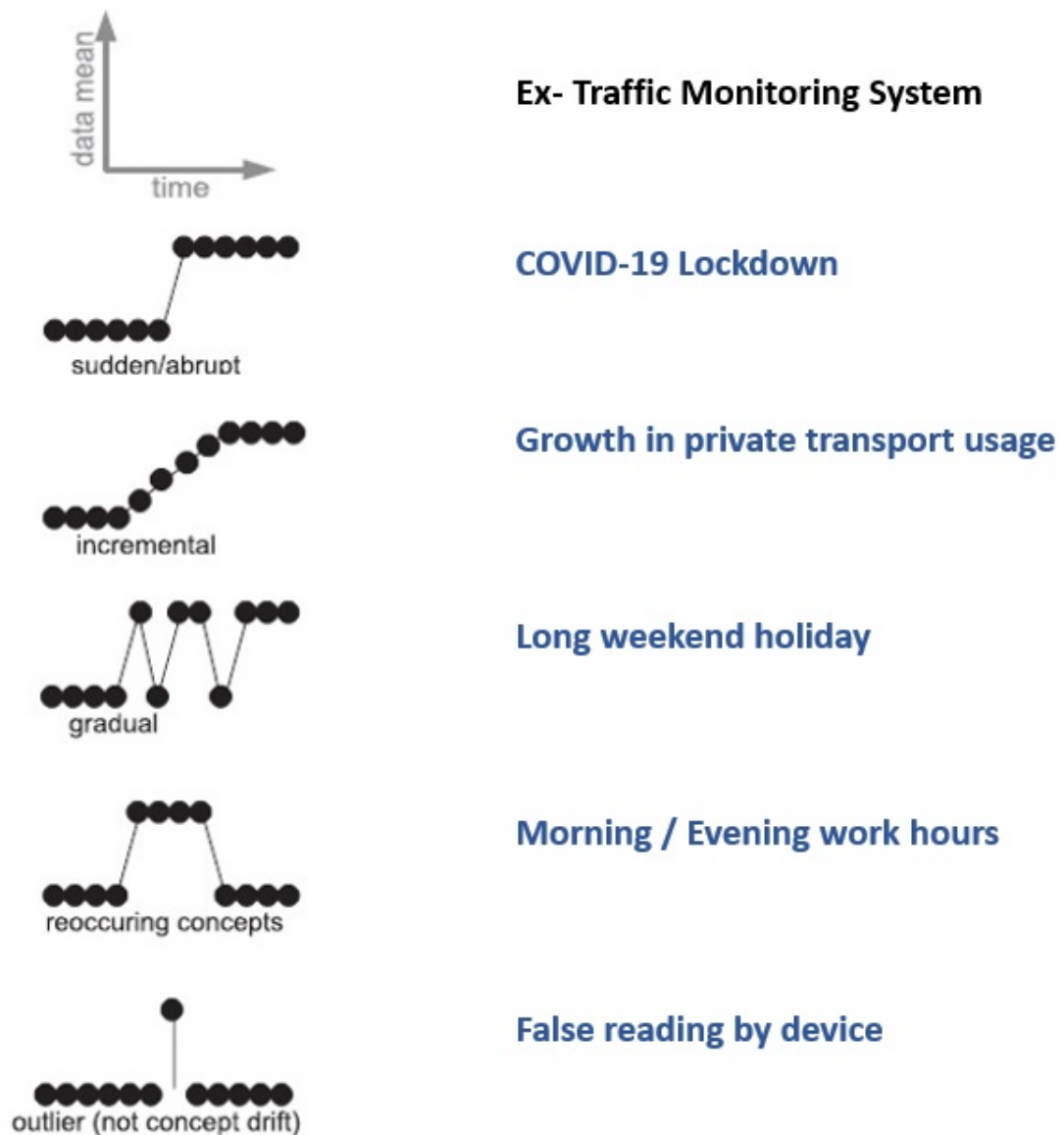


Figure 1.5: Diagrammatic representation portraying different occurrences of data streams.

increasing in an area, causing more population to travel by private transportation rather than public mode, thus witnessing an **"incremental"** increase in traffic. Besides all these types, a false reading by a traffic sensor due to device failure showing an abnormal nature of traffic will not be a concept drift; instead, it would be labeled as an **outlier**.

The spotlight of this study is on the applications, which consist of several heterogeneously distributed data streams. In such a case, even if all the data transmission to a single centralized site is feasible, it would result in misspending of resources if

data could be mined locally (Sajjad, Payberah, Rahimian, Vlassov & Haridi, 2016; Kourtellis, De Francisci Morales, Bifet & Murdopo, 2016). A DDM architecture that allows conjoint processing of these heterogeneous data sources in a streaming context would be an optimal way of processing. Using such an architecture to process the heterogeneously distributed data streams is subject to several constraints. The core problem lies in maintaining a trade-off between accuracy and resources utilized in terms of communication and computational costs. Another problem highlights learning the cross-terms between the heterogeneous distributed sites with minimal resources, and that too in a streaming context, as the sites have no knowledge about each other's data view. However, at the current stage, this research area is immature and requires further endeavor.

1.3 Motivation

A rapid growth in the number of organizations operating globally has turned distributed data processing into a necessity. It is believed that "Data is the new oil of the Digital Economy and data in the 21st century is like oil in the 18th century" (Toonders, 2014). This opinion highlights the importance of data, especially refined data. Therefore, with data, data mining algorithms are equally becoming valuable assets. Moreover, the broadening scale of usage of sensor networks has revived distributed stream mining (Dong, Zhuang, Huang & Fu, 2009; Alaerjan, 2023) (to find the definition of streams, please refer to Section 1.1), emphasizing Heterogeneous Distributed Data Stream Mining (HDDSM). There are tremendous real-world scenarios that utilize different Internet of Things (IoT) data streams and have focused on the future demand for real-time utilization of IoT data streams instead of offline analysis. Social media platforms tend to process the massive volume of data across the globe quickly and efficiently to take effect as immediately as possible, like real-time post monitoring on social media

applications. According to Vuleta (2021), Google, Facebook, Microsoft, and Amazon store at least 1,200 petabytes of information.

This research is motivated by three key reasons. *Firstly*, the global rise in the data generated from distributed streams requires specialized algorithms for online DDM. In some of the distributed networks, there is a scarcity of computational and communication resources. Besides this, due to restrictions on computational resources, IoT devices typically depend on centralized platforms hosted on the cloud for performing intricate analytical tasks such as aggregation, classification, pattern detection, and prediction in stream processing, which ultimately results in issues like connection loss, high response time, single point failure, and overhead in the centralized computing system (Yasumoto, Yamaguchi & Shigeno, 2016). IoT devices are heterogeneous in hardware architecture and limited in processor, memory, and communication capabilities. Also, performing data processing at the edge presents its own challenges due to the heterogeneity of devices, resource constraints, a diverse range of programming languages, and the absence of standardized protocols (M. M. Gomes, da Rosa Righi, André da Costa & Griebler, 2021).

Classification is a major area of interest within the field of HDDSM. The growing significance of classification applications has captivated the attention of researchers (C. Aggarwal, 2007; Kesavaraj & Sukumaran, 2013). The prominence of classification problem in the streaming context has been revealed by a multitude of applications in diverse areas, to mention a few; the real-time monitoring of vital signs data such as heart rate, respiratory rate, temperature, and others generated from IoT-enabled wearable devices can help in classifying abnormalities and early detection of life-threatening conditions. Another example can be an intelligent security surveillance system that uses input from cameras with different features and generates triggers in the event of the detection of unusual behavior, thereby enabling proactive actions.

Nonetheless, most existing approaches for classifying vertically distributed data emphasize centralized learning. Such an approach would not provide an optimal solution for online classification with distributed streams. However, more recent distributed architectures using an ensemble algorithm centralize only the local classification generated from local models into the final classification result (Devi, 2014; Museba, Nelwamondo & Ouahada, 2021). This approach can also be applied to classify data streams as it centralizes only local classification rather than centralizing all the data features. As such, this ensemble modeling architecture doesn't involve much communication cost but compromises classification accuracy. Despite their applicability, these techniques fail to capture the cross-terms that may exist between vertical data streams. Communication cost pertains to the transmission or exchange of records between sites, and a higher communication cost corresponds to an increased transmission or communication of records among sites. The criteria used for measuring accuracy is evaluating model effectiveness (usually the final prediction model) by comparing predicted and true values. For this study, an online classification accuracy calculating environment has been implemented. Further elaboration on these evaluation metrics has been presented in Section 3.4.2.

The *second reason* is the integration of multiple systems or data streams. There have been burgeoning IoT applications covering almost every sector, like industry, healthcare, agriculture, and others. The contributions made by such applications are enabling intelligent city initiatives all over the world (Adi, Anwar, Baig & Zeadally, 2020) by upgrading public transportation, refining population health management, and including many more services. IoT data is currently underutilized by restricting its usage to a specific application, but a broader application that can fuse more data streams by learning relationships between diverse data sources may provide magnificent potential for a range of applications. For instance, data streams from weather, wind, and social media can be correlated to the traffic data stream and can be used to predict the level of

traffic, which proves to be a more efficient and reliable solution (Akbar et al., 2018). While executing these applications, the most crucial question is the conjoint processing of multiple data streams (Sarkar, Nambi, Prasad & Biswas, 2014; Akbar et al., 2018).

Finally, the research on distributed data stream mining has recently received much attention; these applications benefit from analyzing data streams originating from different data sources (Akbar et al., 2018). For example, a real-time monitoring service is used to detect an anomaly in the case of seniors living alone. Such a service uses data from a sensor attached to devices like toilet, refrigerator, and electric pots and detect anomalies like falls in real time and timely inform caretakers and families. Predictive modeling that uses machine learning and data mining to predict and forecast likely future outcomes by analyzing current and historical data has taken place in almost all domains, such as economies, banking, surveillance, and healthcare. The requisite components for any predictive models are reliable data from varied sources, information, and appropriate analytical tools. We have chosen the health sector as the case study for our research to assess our proposed algorithms.

Health is becoming a crucial issue in predictive analytics and focuses on preventing diseases commonly prevalent in society. Health forecasting, which includes risk prediction modeling, is a data-rich field that serves as a valuable tool for predicting health events or situations and forewarning future events. It is aimed towards facilitating health care service provision at the community level. The knowledge gained from implementing such a model could potentially provide additional risk stratification for more efficiency and timeliness in healthcare resource deployment. *Asthma* is one of the most prevalent and costly chronic diseases which cannot be cured. It affects more than 339 million people around the world (Bradley, 2022). Respiratory disease is NZ's third most common cause of death (Barnard & Zhang, 2019). Asthma Respiratory Foundation NZ (2021) states that respiratory diseases caused 87,500 admissions in 2019 (1,780 per 100,000) in NZ. According to researchers, genetics, allergies, infections,

and environmental exposures are all risk factors contributing to asthma presentations to the hospital in any population. On top of this, people with moderate to severe or uncontrolled asthma are more likely to experience severe COVID-19 symptoms or be hospitalized. Temperature, one of the most crucial factors affecting asthma exacerbation, change variably in Auckland, NZ (NIWA, 2006; Barnard & Zhang, 2021). Two consecutive days may have very different readings for temperature, thus affecting an asthma patient differently. Therefore, a prediction model that can predict the demands for ED due to asthma patients using data from heterogeneous sources³ in near real-time can provide probable productivity to health care providers and organizations in predicting events before they occur, allowing early medical intervention.

1.4 Problem Statement

A conceptual framework (Miles & Huberman, 1994) for this study has been illustrated in Figure 1.6, which explains the correspondence between an area of emphasis to the complete frame of reference for data mining. The figure consists of three layers: *context*, *focus area*, and *case study*. The *context* lays the framework for this work while *focus area* and *case study* would help in stating the problems that would take the shape of research questions.

The context includes *data mining*, *data streams*, *distributed data*, and *supervised learning*. *Data mining* is the process of discovering significant, inferred, previously unknown, potentially valuable, and understandable patterns and correlations from huge datasets (Chung & Gray, 1999; Hand, Smyth & Mannila, 2001; Bertoni & Larsson, 2017; Qingqing Chang, 2022). In light of most real-world applications, data exists in a *distributed fashion*, which may be partitioned horizontally or vertically over the sites. Mining of *data stream*, which involves a continuous flow of data, has earned its

³Contribute to parameters that affect asthma conditions

place in widespread applications ranging from time-critical astrophysical applications to the online decision-making process in business and trades (Mohamed, 2010; Ilin & Klimin, 2019). *Supervised learning* algorithms (Q. Liu & Wu, 2012) require that the prediction of the unknown target feature value be computed as soon it is received, thus enabling real-time analysis (G. Mao & Qiao, 2017). An example that can be cited here is a congestion predictive model (Sarker, 2021). It analyzes streams of data originating from traffic sensors and determines the status of congestion, classifying it into a) a state of congestion or b) no congestion.

The focus area highlights the key areas and includes *vertically distributed data streams, classification, distributed processing, and cross-terms*. *Classification* is the most commonly used supervised data mining technique to classify a dataset in one or more pre-defined classes. Classification in a *vertically distributed* scenario where data originates from heterogeneous distributed streams is the focused area of this study. Besides privacy and security concerns, each site would not prefer to share its data. Moreover, aggregating entire data from all the sites to a central site experiences several limitations, such as the involvement of enormous resources in communication and computation. Existing approaches for classifying vertically distributed streams suggest the centralization of data or construction of a centralized classifier at a central site, which means that for classifying a record, all of its features must be available at the location of centralized classification (R. Chen, Sivakumar & Kargupta, 2001; Kourtellis et al., 2016). A more generic approach to this scenario is a *distributed processing* architecture that aggregates local classification generated from local models using some ensemble learning technique (Rokach, 2010; Museba et al., 2021). However, this architecture does not require much data transmission but fails to capture the *cross-terms* between the distributed sites, thus witnessing a sharp decline in accuracy.

A primary concern in processing heterogeneous distributed sources is maintaining the *trade-off between the accuracy of classification prediction and resources utilized*,

which produces **Research Question RQ1**. This question aims to optimize resource utilization in vertically distributed data streams while maintaining similar accuracy levels. Some records require processing from a combined perspective to produce accurate classification results, which requires capturing relationships as cross-terms between the vertically distributed streams. Learning cross-terms between sites requires the transmission of data from sites to a central location, raising concerns. Thus, in **Research Question RQ2**, we introduce a mechanism for learning the cross-terms between the sites with minimum data utilization.

The innermost layer defines the *case study* which was used for testing the performance of algorithms proposed as solution to RQ1 and RQ2. Organization for Economic Co-operation and Development (OECD) statistics indicate NZ has one of the highest hospital admission rates for asthma among OECD countries. Not much work has been accomplished in population health forecasting for asthma in NZ. Asthma is a condition that is affected by heterogeneous factors; thus, generating a model leveraging diverse data sources to predict peak demand days for asthma-related ED visits and admissions at the population level would provide timely and targeted influences to minimize the socio-economic impact of asthma. **Research Question RQ3** focuses on developing predictive modeling for asthma using data from diverse sources in centralized offline architecture and further executing it on DDM architecture in a streaming context.

1.5 Research Questions

Heterogeneous distributed data sources are the base of this research. Streaming heterogeneous data sources that produce distributed data streams are increasingly prevalent in various environments. Some examples include IoT devices, sensor networks, social media platforms, web data streams, satellite imagery providers, etc. (Parthasarathy, Ghoting & Otey, 2007), which represent true vertically distributed data mining scenarios.

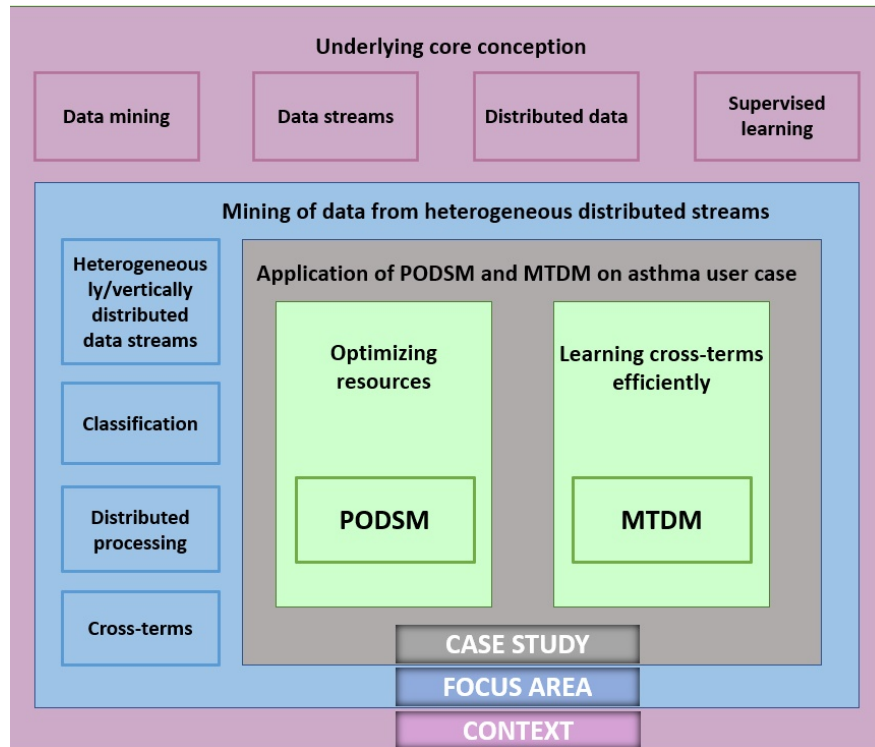


Figure 1.6: A conceptual framework depicting the domain of interest in the frame of reference.

We will be using classification as the data mining technique; therefore, the classification of data from heterogeneous sources (or vertically distributed data) forms the core for the rest of the work. As stated earlier, the stimulus in selecting the classification problem (C. C. Aggarwal, Han, Wang & Yu, 2004; Sarker, 2021) in this research concerns their popularity as an extensively studied subject (Kesavaraj & Sukumaran, 2013). Research questions RQ1 and RQ2 have been prepared for a resource-constrained distributed data stream environment, while in research question RQ3, we use the asthma case to validate the application of the algorithms developed as part of RQ1 and RQ2. Each research question and its associated inner queries have been summarized for further study and experimentation.

RQ1 What role does past data play in reducing the communication overhead in vertically distributed data stream mining?

RQ1.1 Is there any way the confidence of other distributed sites can be estimated

with no knowledge about its view?

RQ1.2 Does any attempt to decrease communication affect the accuracy of classification?

RQ1.3 Can we achieve savings in resource time by utilizing such a methodology?

RQ2 Does an approach involving minimum centralization of records facilitate easy learning of the relationship between the distributed sites?

RQ2.1 Is this method better in performance than the state-of-the-art brute force strategy

RQ2.2 Is this method scalable and robust concerning the increase in the number of distributed sites?

RQ2.3 How efficient is this method in maintaining the balance between accuracy, communication, and resource time?

RQ3 What benefits can be achieved by applying methodologies devised in RQ1 and RQ2 on predictive modeling for asthma?

RQ3.1 Which heterogeneous distributed data sources are taken advantage of for generating a predictive model for asthma?

RQ3.2 In what ways does employing diverse heterogeneous sources affect the performance of the prediction model?

RQ3.3 Can comparable capabilities similar to a centralized architecture be accomplished by employing a decentralized model?

RQ3.4 What effect does an online learning paradigm have on the asthma prediction model's performance?

1.6 Research Objectives

The objective of this research is to propose a suite of algorithms that can be targeted towards mining data from heterogeneous sources, particularly in resource-constrained scenarios. We would use a hierarchical distributed structure to mine data from vertically distributed sites, allowing conjoint processing of these heterogeneous data sources. The output received from this work would enhance the efficiency in mining data from resource-constrained heterogeneous distributed data sources. This research would produce novel solutions for heterogeneous data stream classification and interest many domains, e.g., security and sensor network applications. Thus, these solutions would be applicable in diverse areas. The objectives can be summarized as follows:

1. Proposing a novel algorithm; *Performance Optimizer in Distributed Stream Mining (PODSM)*.
 - A data transmission protocol that reduces the data transmission volume in a DDM context while maintaining similar accuracy levels.
 - Demonstrating the capability of the algorithm by achieving significant improvements in resource time.
 - Augmenting the proficiency of local models by building a feedback system that helps the distributed sites in making predictions about the confidence status of their peer sites.
2. Devising a second novel algorithm, *Minimized Tree for Distributed Mining (MTDM)*.
 - A novel method that eases the procedure of learning relationships between vertically distributed sites.
 - Exhibiting the algorithm's potential to maintain the accuracy levels.

- Resource time improvement in determining the cross-terms between vertically partitioned sites.
3. *Performing predictive analysis* for the asthma case study using real datasets from heterogeneous data sources.
- Exploring the nature of the correlation between various data sources and asthma-related admissions.
 - Devising a staged framework for obtaining a population-based predictive model to predict peak demand days for asthma-related hospitalizations in Auckland, NZ.
 - Exhibiting the usability of Google search trends in health forecasting models for NZ.
 - Developing an offline centralized prediction model for asthma using standard machine learning algorithms.
4. Testing algorithms, *PODSM* and *MTDM* on the asthma case study.
- Utilizing decentralized architectures in the streaming mode for executing predictive modeling for asthma.
 - Executing algorithm *PODSM* and *MTDM* for optimizing performance in the asthma prediction model and assessing their comparison with state-of-the-art methods.
 - Demonstrating the proposed algorithms' (*PODSM* and *MTDM*) potential to dynamically trade-off accuracy with computational resource time and communication overhead in this actual case study.

Figure 1.7 showcases the relationship between research objectives and research questions, providing a clear and visual representation of how this study addresses the

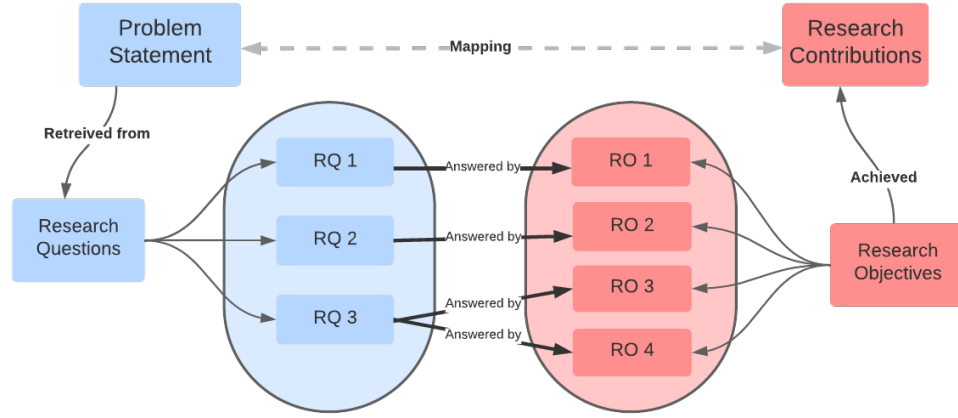


Figure 1.7: Visual representation illustrating the mapping of research questions to research objectives.

research objectives.

1.7 Thesis Contributions

An outline of the research solutions proposed in this study addressing the research problems mentioned in Section 1.4 with respect to HDDSM has been depicted in Figure 1.8. The outcome of this work is composed of two novel algorithms: *PODSM* and *MTDM*, that can be exploited to achieve specific goals in resource-constrained scenarios. In the next section, the applicability of PODSM and MTDM is verified in a real-world case study. The goals achieved in the proposed solution target resource conservation in the online classification of heterogeneous distributed data streams.

An initiative to realize the research objectives mentioned in Section 1.6 has facilitated the contribution of definite solutions discussed below.

1. An algorithm called Performance Optimizer in Distributed Stream Mining (PODSM) exploits the modified version of Bayes' theorem for saving resources in mining data from heterogeneous distributed data streams (Chapter 3).
2. A tree-based algorithm known as Minimized Tree for Distributed Mining (MTDM)

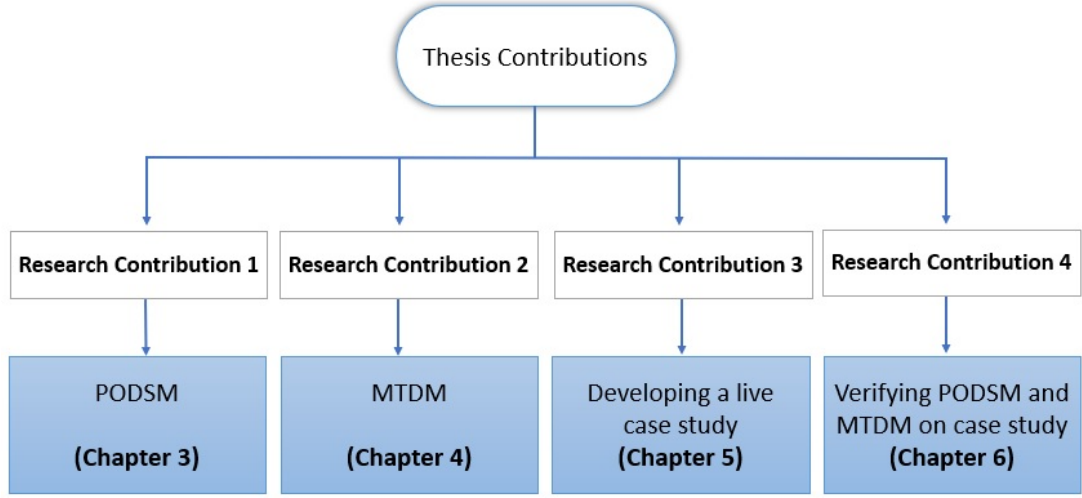


Figure 1.8: A sketch representing an overview of the research contributions and their mapping to the thesis in the form of chapters.

serves as an efficient and scalable approach using a tree to learn the cross-terms between heterogeneous or vertically distributed sites (Chapter 4).

3. A staged framework for conducting predictive analysis of a real-world case study using the real data from real-world heterogeneous distributed sites has been implemented (Chapter 5). This phase also included the execution of standard machine learning algorithms in offline mode.
4. To test the applicability of algorithms, PODSM, and MTDM in a real-world application, the aforementioned algorithms were implemented in the case study studied through Objective 3. This phase was to implement the online distributed architecture and check the proposed algorithms' (PODSM and MTDM) performance in this case study (Chapter 6).

The outcome achieved from this work has produced the following publications:

1. **Bhalla, Rashi** & Pears, R. & Naeem, Muhammad Asif & Mirza, Farhaan. (2022). Novel method for optimizing performance in resource constrained distributed

data streams. *Applied Intelligence*. <https://doi.org/10.1007/s10489-021-03019-5> (Chapter 3).

2. **Bhalla, Rashi** & Pears, R. & Naeem, Muhammad Asif & Mirza, Farhaan. (2021). Scalable and Efficient Method for Learning the Relationship between Heterogeneous Distributed Data Streams Using a Minimized Tree. Under review in *Data Mining and Knowledge Discovery* (Chapter 4).
3. **Bhalla, Rashi** & Mirza, Farhaan & Naeem, Muhammad Asif & Chan, Amy Hai Yan. (2022). Predicting Peak Demand Days for Asthma-Related Emergency Hospitalisations. Under review in *BMC Medical Informatics and Decision Making* (Chapter 5).
4. **Bhalla, Rashi** & Mirza, Farhaan & Naeem, Muhammad Asif. (2022). Online Processing of Heterogeneous Data Sources for Predicting Peak Demand Days for Asthma-Related Emergency Hospitalizations. Under review in *Machine Learning* (Chapter 6).

1.8 Outline of the Thesis

The rest of the thesis has been organized as follows:

Chapter 2 establishes a bedrock for this work. It is broadly divided into two parts: the first focuses on DDM with major significance in scenarios where data is heterogeneously distributed, particularly in a streaming context, and the second on the asthma case study. Both of these sub-parts reviews the previous work done in their respective area. Also, it provides a thorough discussion about the foundation work for this study. The chapter concludes by analyzing the research gaps in recent work.

Chapter 3 presents a novel algorithm, PODSM, designed to optimize performance in mining data from distributed streams. The main motive behind this algorithm is

to gain savings in terms of communication and resource time without compromising accuracy. The efficacy of this algorithm has been demonstrated using experimental evaluation and comparison with state-of-the-art methods on 11 benchmark datasets.

Chapter 4 exhibits another novel algorithm, MTDM, devised for a heterogeneous distributed scenario. It reveals a scalable and robust method for learning the relationship between the distributed sites, which ultimately leads to economizing resource time. As such, it demonstrates the potential of trees in learning the cross-terms. The performance is evaluated by executing the algorithm on 11 machine learning benchmark datasets. Also, it provides a detailed analysis of the findings.

Chapter 5 considers the generation of a real application for verifying the implication of PODSM and MTDM. This chapter discusses the thoroughgoing agenda used for building the real dataset for asthma which is ultimately used as a use case. In subsequent work, an offline centralized prediction model that can predict peak demand for asthma-related emergency hospitalizations was produced using standard machine learning techniques.

Chapter 6 is dedicated to generating the experimental comparison between a centralized and decentralized architecture in online mode. Various experiments are executed towards assessing different algorithms under decentralized architecture in the asthma case study. The savings gained in terms of resources and the classification accuracy achieved by decentralized architecture are analyzed. Finally, algorithms PODSM and MTDM are tested, and their execution procurement is summarized.

Chapter 7 summarizes the thesis by featuring accomplishments and key outcomes. Moreover, it discusses the implications of the novel algorithms proposed in this work and records the limitations imposed. Besides this, additional work has been done discussing variations of the present work that can be used as the basis for further research and provides directions for future scope. The conceptual roadmap for this thesis has been presented in Figure 1.9

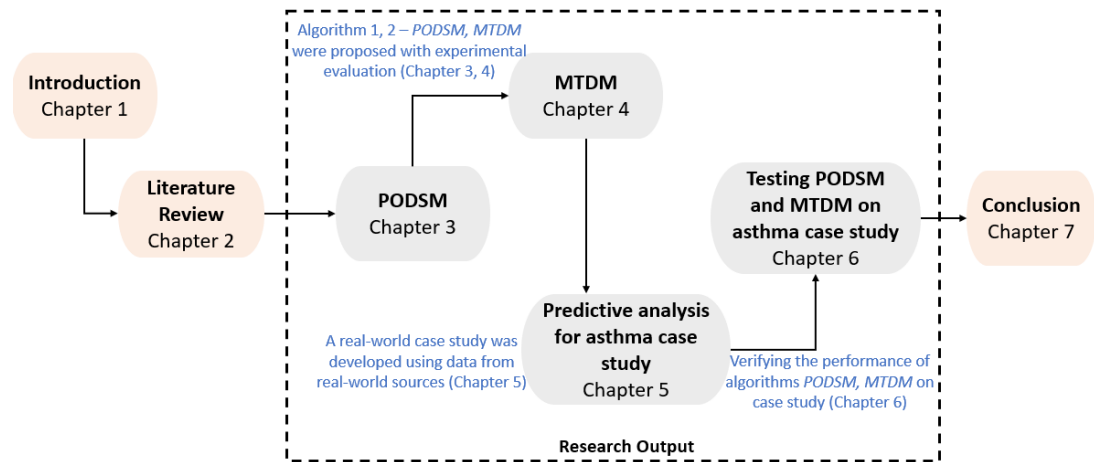
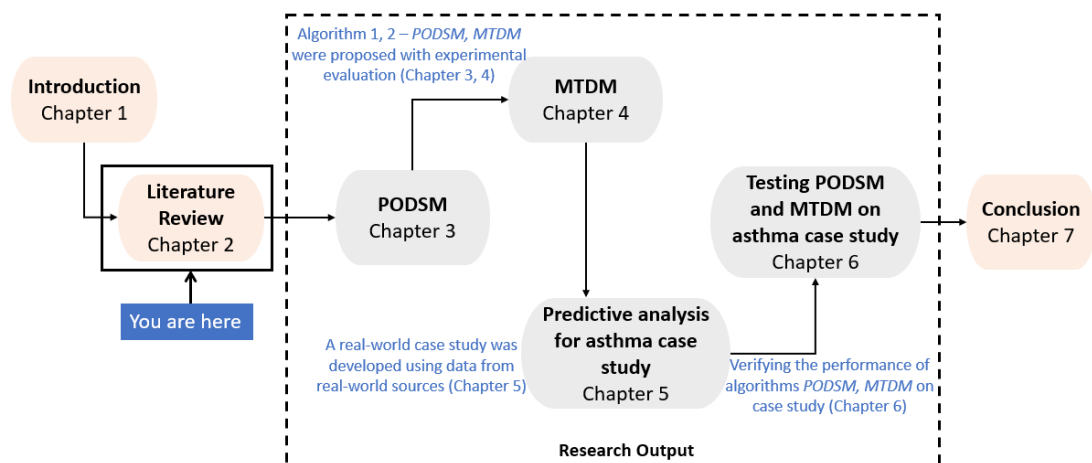


Figure 1.9: A roadmap demonstrating the structure of the thesis.

Chapter 2

Literature Review

This chapter aims to examine a detailed overview of the state-of-the-art studies to develop a thorough understanding of mining heterogeneous distributed data streams and predictive modeling for asthma. The literature review has two key parts: the first begins by laying out the theoretical background of the research topic, and it will go on to review recent studies into distributed stream mining with an emphasis on the heterogeneous arena. The goal of the second part is to trace the background knowledge and work done towards analyzing parameters that affect asthma conditions and methods undertaken to generate predictive modeling for asthma.



Roadmap of the thesis.

2.1 Introduction

The present chapter presents an encapsulation of the related literature that forms the framework of this research work. The discussion commences by exploring various aspects of *DDM*, *data streams*, *heterogeneous distributed data streams*, and *asthma prediction models*, as they constitute the fundamental elements of this study. Thereafter, the challenges experienced in mining data from distributed sources, specifically in the streaming context, are explored. Work done in connection with the prediction models developed for asthma has also been investigated. This is vital for understanding the state-of-the-art of data mining in the asthma prediction model and identifying the knowledge gaps, particularly pertaining to models that aim to predict peak demands for asthma-related hospitalizations. It would particularly help build a live case study and test the proposed algorithms' applicability by executing and verifying their performance on the actual dataset. Fundamentally, this research addresses the problem of HDDSM in resource-constrained scenarios; expressly, we propose a suite of algorithms that emphasizes saving resources without compromising accuracy. Therefore, approaches related to DDM, data stream mining, and, most importantly, HDDSM are crucial in this study.

In order to address issues related to mining data from heterogeneously distributed data streams, different methods have been devised by the researchers. Moreover, a plethora of work has been done concentrating on data analysis and prediction modeling for asthma. Therefore, to bring out the work done by other researchers in the domains presented in Figure 2.1, the fundamental research interest has been indicated below:

1. Distributed Data Mining in Offline Context
2. Data Stream Mining
3. Heterogeneous Distributed Data Stream Mining

4. Predictive Modeling via Heterogeneous Sources for Asthma

5. Background Work

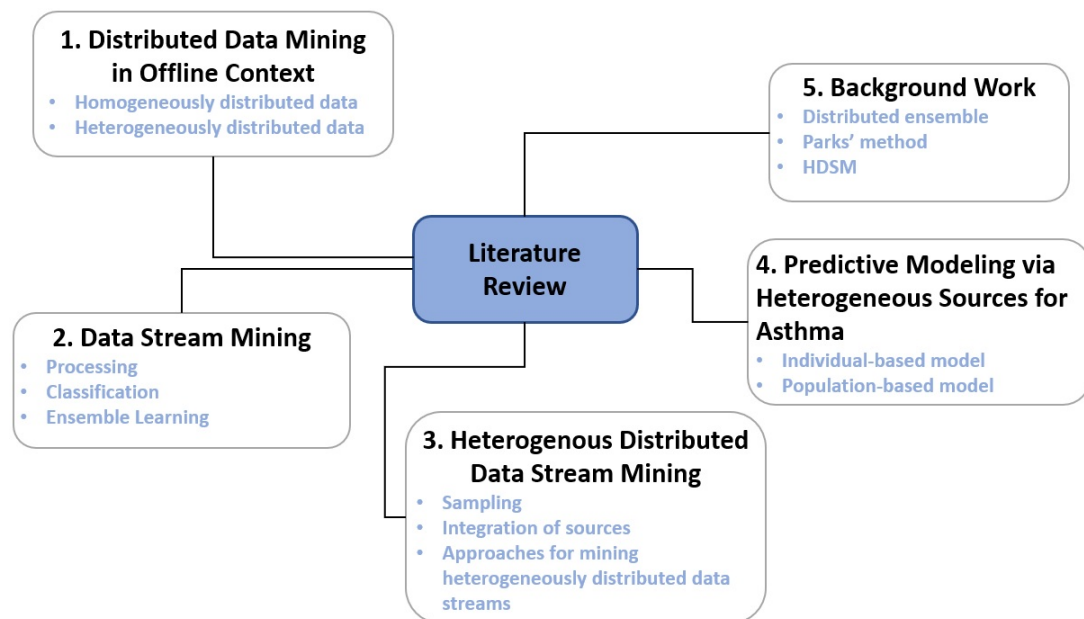


Figure 2.1: Sub division of literature review.

The initial two sections start with a broader scope which narrows down in the third section. The first section covers studies undertaken in the field of distributed (heterogeneous/homogeneous) data mining in an offline context. Whereas the work conducted in the field of data stream mining, especially in the classification arena, has been elaborated upon in the second section. The third section explores research that focuses both on heterogeneous distributed aspects and stream mining. Additionally, it includes studies that have been extended by adding a streaming mining approach (Denham, 2019) to distributed mining (Park, Ayyagari & Kargupta, 2001). Therefore, it provides an indication that algorithms that are capable of mining distributed data can be widened for streaming context by devising and adding a methodology to serve fast-arriving huge volumes of data. The next section features the predictive analysis and modeling for the asthma case study. Lastly, the background work gives a summary of

the state-of-the-art and most comparative foundation work related to the work discussed in this thesis. The studies reviewed in these sections facilitated a thorough gap analysis, enabling us to pinpoint specific areas where research gaps exist.

2.2 Distributed Data Mining in Offline Context

Nowadays, the volume of data generated is incredibly increasing in organizations (Ilin & Klimin, 2019; Mikalef et al., 2019; Vuleta, 2021). The non-trivial question that arises is how to interpret this enormous data and acquire useful information and knowledge. The clear answer to this query would be *data mining*, a field that intersects Computer Science and Statistics (Agarwal, 2013). Data mining creates fertile ground for implementing new techniques and methods to obtain more transparent, informed, and autonomous decision-making (L. D. Chen, Sakaguchi & Frolick, 2000; Bertoni & Larsson, 2017). Nevertheless, the process of data mining faces several challenges (Q. Yang & Wu, 2006; Y. Zhou, Tian, Liu, Liu & Zhu, 2019), with one of them being mining distributed data.

Rapid technological advances have triggered the prevalence of wide-ranging distributed computing environments. The majority of these scenarios witness a vast volume of data generation from distributed nodes (K. Liu, Kargupta & Ryan, 2004). Currently, in almost every application domain, there is a dire need to analyze large datasets maintained from various geographically dispersed locations. In 2018, 2.5 *Quintilian bytes* of data were created every day. Mining data from such sources would require data centralization at a data center (Vuleta, 2021).

Nevertheless, accumulating data at a fusion site using a centralized algorithm involves considerable communication among the distributed nodes, which ultimately face complexities aligning in the distributed arena (Gan et al., 2017). Moreover, agglomerating data at a centralized location would be expensive and eventually infeasible due to various reasons like privacy (Clifton, Kantarcioglu, Vaidya, Lin & Zhu, 2002). In

order to cope with these issues, distributed and parallel data mining algorithms were developed (Alhaj Ali, Vařacha, Urbanek, Krayem & Žáček, 2018). In DDM, the same or different mining algorithms are applied at the distributed sites, and global knowledge is obtained by combining the local pattern discovered by applying these algorithms (Kargupta et al., 1999). While a global parallel algorithm is applied to the entire dataset in case of parallel data mining, (Zeng et al., 2012). The algorithms devised for distributed environments can be split into: *efficiency-focused and privacy-focused* (da Silva et al., 2005). The primary goal of the first class of algorithms is to increase efficiency in terms of communication and/or computation (Mallik, Sarda, Kargupta & Bandyopadhyay, 2011), while the second one focuses on preserving privacy (K. Liu, Kargupta & Ryan, 2006).

The basic procedure of DDM can be divided into three steps (Kulkarni, Desai, Ahmed, Vadavi & Yardi, 2007):

1. ***Local knowledge creation:*** Create the local knowledge from the local data available at each distributed node.
2. ***Global knowledge acquisition:*** Obtain the global knowledge by amalgamating the local knowledge.
3. ***Evaluation:*** Checking the performance of the global knowledge.

DDM emerged as an expansion to data mining for distributed data environments. It is a worthwhile subject that demands research (B. Liu, Cao & He, 2011; Zeng, Xu, Shi, Wang & Wu, 2007). Under the DDM trend, either the computational resources or the data is distributed; its application area includes parallel supercomputers, P2P networks, and sensor networks.

Recent technological developments in wireless networks and microelectromechanical systems have contributed to the growth of small-scale, low-cost, and low-power

devices called wireless sensor nodes (X. Chen & Rowe, 2011). They are fundamental Wireless Sensor Networks (WSN) entities and can collect information from the environment, such as temperature, humidity, pressure, vibration, sound, pollutants, and so forth. These nodes communicate their wireless data to the primary location or sink where the data can be observed and analyzed. The sensor nodes possess the power of sensing, processing, storage, and communication and are thereby deployed in large numbers over vast geographical regions to serve various applications like military, health, environment, and transportation, among others (Matin & Islam, 2012). A great deal of previous research into distributed mining has focused on sensor networks (Cantoni, Lombardi & Lombardi, 2006; Mahmood, Shi, Khatoon & Xiao, 2013; Ma et al., 2004).

Mohanty, Lydia, Elhoseny, Otaibi and Shankar (2020) have proposed a model based on Recurrent Neural Network - Long Short Term Memory, abbreviated as RNN-LSTM, to obtain competent and ideal balancing of the load in WSN. The goal behind this work was to reduce the load at the fusion site and minimize the energy consumption in the network. The network is divided into several layers and placed into the sensor nodes. Therefore, this work targets reducing the overhead at the fusion center by transmitting processed data by the RNN-LSTM model rather than sending the actual data. In another work by Brooks, Ramanathan and Sayeed (2003), a Collaborative Signal Processing model for efficient target classification and tracking in sensor networks was proposed. It was based on location-aware data routing involving only an important node subgroup, thus utilizing limited communication and computation abilities. Even though the centralized architecture is feasible for tracking in sensor networks, it brings several loopholes: *delayed throughput, inefficiency concerning energy and bandwidth, synchronization problems, and single point of failure* (Carlos-Mancilla, López-Mellado & Siller, 2016). Hence, a distributed tracking approach was identified by Brooks et al. (2003) that can implement multiple threads simultaneously in a peer-to-peer network. Moreover, for classifying the target, this method uses fusion techniques, following the

distributed implementation in the sensor network.

Villaverde, Aledo, Pérez and Moreno (2019) put forward a two-staged structure for examining the data collected from various sensor nodes. In the first stage, the base models are trained using the Artificial Neural Network (ANN) on different sets of data using their partial view of the entire data and obtaining their classification estimates. In the next stage, the predictions are combined using four different ensemble methods; plurality majority, basic weighted majority, and stochastic weighted majority. Ensemble learning has been implemented in a broad domain; Pintelas and Livieris (2020) have discussed various ensemble methods and listed a multidisciplinary range of applications where ensemble methods find their place.

A typical distributed mining scenario can fall into any of the following categories:

1. ***Centered clusters/peer-to-peer:*** (Leibnitz, Hoßfeld, Wakamiya & Murata, 2007)
Under *centered clusters*, there is a coordinator that partitions the work into multiple computers and maintains cooperativeness. *Peer-to-peer* does not involve a central server and is decentralized in nature; each site receives the data and works on its own task.
2. ***Single model/meta-learning:*** (Tsoumakas & Vlahavas, 2002) Under *single model*, a data mining algorithm is divided and assigned to each distributed site. The local learning model is generated at each site, which are then communicated to the global location whilst *meta-learning*, referred to as learning about learning, represents the algorithms that learn from outcomes of other machine learning algorithms which are trained on data present at distributed sites.
3. ***Homogeneous/heterogeneous:*** (Y. Liu, Xu & Li, 2018) This group suggests partitioning of data rather than resources. Suppose a network is composed of J nodes spread over a geographic region. For a schema consisting of L data samples (or records) and M attributes, in the *homogeneous approach*, each node consists

of M attributes and a subset of L data samples, represented as $X_{l,M}^j$ where $l \in L$ and $j = 1, 2, \dots, J$, on the other hand under *heterogeneous partitioning*, each node is rendered as $X_{L,m}^j$ where $m \in M$ and $j = 1, 2, \dots, J$. In heterogeneous data distributions, all the local sites witnesses differing views, and the main challenge lies in learning the global model from these local models (Tsoumakas & Vlahavas, 2002).

In light of recent events in IoTs and sensor networks, it is becoming extremely difficult to ignore the existence of data distribution. Thus, in this research work, we are considering a distributed scenario where the data is distributed across the sites. Henceforth, we will be leaning towards work where data distribution forms the prime subject. The section below exhibits algorithms that have been proposed for i) *homogeneously distributed data*, ii) *heterogeneously distributed data*, and iii) *homogeneous or heterogeneous distributed data*.

2.2.1 Approaches for homogeneously distributed data

In a distributed scenario, the prevailing method for data collection is characterized by homogeneity, wherein the features that describe the data remain consistent across all distributed sites. This section examines the algorithms for homogeneously distributed data mining. Weinberg and Last (2017) proposed a novel approach called Syntactic Similarity Method, also known as SySM, for large distributed data environments. The presented framework deals with decision trees constructed by a horizontal partitioning approach, in which the researchers came up with the idea of choosing one representative model from multiple trees, unlike an ensemble approach (Bühlmann, 2012), which combines the predictions from multiple decision trees. The classification accuracy of SySM is on par with ensemble majority voting (Rokach, 2010). Their methodology has its basis on the assumption that the two trees that produce the same predictions are due

to their internal structure being similar.

Y. Yang et al. (2019) used a distributed algorithm to aggregate and predict power generation and consumption over a distributed smart grid. The nature of data is assumed to be homogeneously distributed (Y. Liu et al., 2018) at the different sites. The empirical study showed that the communication cost increased linearly with a centralized algorithm, while for a distributed algorithm, the increase was logarithmic. On the other hand, with the distributed approach in place, a small degree of error was witnessed. Another work proposed by Datta, Bhaduri, Giannella, Wolff and Kargupta (2006) developed a method for generating clusters in distributed data over P2P networks using K-means. The algorithm used a message-passing technique to maintain the peer connection. It was divided into two steps; in the first step, the peer nodes execute the standard K-means algorithm (Likas, Vlassis & J. Verbeek, 2003), which assigns its data points to the nearest cluster centroids and maintains the necessary information about local cluster centroids, a count of clusters. In the second stage, a poll message is sent to share all the stored information between the peers, according to which the peer nodes update their K cluster centroids by calculating a weighted average of its centroids with all the other centroid information it has received from other nodes.

Y. Li, Bai and Reddy (2016) suggested an approach for mining healthcare data collected from different geographic locations under privacy constraints. This mining framework learns a model for each participating site by approximating the local data distribution. In the next phase, the knowledge gained from local data is transferred in the form of decisions made using the models rather than sharing the patients' sensitive information, thus preserving privacy.

2.2.2 Approaches for heterogeneously distributed data

In the context of DDM, an earlier model that concentrated on heterogeneous databases was Worldwide Relational Learning Daemon (Aronis, Kolluri, Provost & Buchanan, 1997). This system was intended for inductive rule learning from multiple distributed databases. This method worked by labeling each item marker (indicating items in the interest of the concept and its complement), and then these were propagated through several databases looking for values where in or out-of-concept markers accumulate. This process could span several databases that may be located on different machines. However, the drawback was that it relied on manually maintained joins between distributed databases, which could be time and memory-consuming and requires the assumption of a standardized vocabulary across heterogeneous databases.

T. Li, Zhu and Ogihara (2003) have raised a problem that arises while approximating the global model based on the local models, which are generated from different individual databases and differ in characteristics. They have proposed a similarity-based DDM framework to address this case. Moghadam and Ravanmehr (2017) have developed a novel method known as Multi-Agent Hierarchical Data Mining or MHDM for DDM to classify meteorology data collected from sites spread over a vast geographic area of a country (Iran in this case study). This work uses a modified version of the REPTree algorithm over heterogeneously distributed nodes.

An Agent-based Data Mining Info-Structure, a package offering data mining services plus the intelligent agents, was proposed by Zaidi, Abidi and Manickam (2002). The authors have demonstrated its ability to manifold heterogeneous healthcare archives for decision-making and planning of resources. Another agent based model known as Multi-Agent Based Data Mining Aggregation used statistical characteristics of the agents for generating global knowledge (Vijjeswarapu & Natarajan, 2018).

R. Chen, Sivakumar and Kargupta (2004), presented a collective way of forming

a Bayesian Network from distributed heterogeneous data. In this approach, each site produced a local Bayesian Network based on its local view of data. The local sites then identify samples or observations for which the local models are not a good fit. These observations have a low value for local likelihood functions¹ and are expected to indicate the coupling between local and foreign variables available at other sites. As a result, these observations are transmitted to the central site to form a non-local Bayesian Network likely to represent linking between various sites. Finally, a collective Bayesian Network is constructed by merging local and non-local networks.

Another method proposed by Basak and Kothari (2004) builds local classifiers using a fragmentary view of the data available at a local site. These classifiers were built using algorithms like decision trees, neural networks, and k-nearest neighbors. Finally, a classifier is generated using these local classifiers using a classification algorithm to achieve an error rate close to a classifier generated with a complete feature set list. The decision for the final class label is made by computing the overall approximate value of the posterior for a pattern or a data row using only the set of consistent classifiers. Hence, the final outcome is determined by selecting the class label with the maximum value of $\hat{P}(w_i | x)$, where w_i represents the class label, x denotes the pattern, and $\hat{P}$ represents the posterior probability.

2.2.3 Approaches applicable to both distribution of data

As discussed earlier, the distributed data can be mined in two ways; *one* by combining the sole models generated at geographically distributed sites and *second* by building a central model with all the data (Agrawal, Imieliński & Swami, 1993; Han, Pei & Yin, 2000). The centralized model generated with the entire dataset is more accurate and expensive regarding resources when compared to a distributed architecture. An

¹A likelihood method provides an estimate about how aptly a particular model can fit the data.

algorithm known as OPTimal Data and Model Partition (Turinsky & Grossman, 2000) was developed to find an intermediate strategy that bridges centralized and distributed mining. The motive was to formulate a mathematical programming problem that minimizes the cost function subject to error constraint, achieving a comparable accuracy level to a centralized model with the minimum possible cost.

The call in question in the case of DDM is to learn an ample amount of knowledge from distributed databases equivalent to a centralized architecture, enforcing a cutback on the amount of resources utilized. H. Qi, Wang and Birdwell (2004) have demonstrated a distributed Principal Component Analysis algorithm that derives the global principal component by integrating local covariance matrices. The work proves that the global principal component derived for homogeneous databases is the same as the centralized one, but an error is introduced while recompiling the component for heterogeneous databases. The authors discussed in depth the two approaches used for distributed mining, the first being the centralized approach of mining prior to any processing. The second is targeted towards generating a global model from partial knowledge learned from the local models. How to integrate partial knowledge to obtain the global model is a serious issue (Tsoumakas & Vlahavas, 2002; Dong et al., 2009).

Kholod et al. (2019) have discussed an approach to parallelizing the process of training the Naive Bayes Classifier on the distributed data. This method is based on Bayes' theorem and can be applied to both vertically and horizontally partitioned data. The most complicated case of vertical distribution is when each site has only one attribute with duplication of the target attribute. Skillicorn and McConnell (2008) have proposed a methodology for attribute ensemble in vertically partitioned distributed data where local models were built using decision trees or neural networks with a single attribute and were subsequently tested to determine their quality. Assuming that the new records arrive in a similar distributed fashion, the local models send their prediction results to the central site, which were then used as candidates to generate the final prediction. For

a classification problem, majority voting was applied, while averaging was done for regression (Woods, Kegelmeyer & Bowyer, 1997). Skillicorn and McConnell (2008) have revealed the competency of the algorithm through experimentation for both horizontally and vertically partitioned data.

Work presented by Ye et al. (2019) addressed the security concerns by proposing the privacy-preserving version of the C4.5 algorithm that can be applied to both homogeneous and heterogeneous partitioned data. The work uses two outsourced protocols for ensuring privacy in the DDM structure: Outsourced Privacy Preserving Weighted Average Protocol or OPPWAP and Outsourced Secure Set Intersection Protocol or OSSIP. Schemes for preserving privacy have been extensively studied by Zhao et al. (2018) for regression and binary classification (Ganaie, Hu, Malik, Tanveer & Suganthan, 2022).

2.2.4 Challenges in DDM

Most organizations are currently following the DDM approach for working in different areas. A number of enterprises make use of data mining for Business Intelligence (Hsu & Wallace, 2007; J. Luo, Xu, Jamont, Zeng & Shi, 2007). Manohar, Jenifer, Nisha and Benita (2021) identified a method to analyze customer churn by merging the result generated after applying techniques like Bayesian Networks, Support Vector Machines (SVM), and Random Forest (RF) using a collective approach. There are tremendous other beneficial areas of operation of DDM; credit card fraud detection (Chan, Fan, Prodromidis & Stolfo, 1999), air pollution monitoring sensor networks (Khedo, Perseedoss & Mungur, 2010), and so forth. The main advantages of using DDM (Fu, 2001; Roughan & Zhang, 2006) are summarized below:

- **Reduced Network traffic:** With DDM, the distributed sites send only the results from the local models to the fusion node, drastically reducing the transmission volume compared to a case where the entire data is communicated. Additionally,

it aids in reducing resource time as well.

- **Enhanced security system:** With a method in place to transmit only the results instead of data, the organization is safer by offering a better privacy solution.
- **Improved Performance:** Multiple queries can be executed simultaneously on different sites, leading to performance improvement.
- **Capability to handle large datasets:** DDM can handle large datasets that are beyond the capability of data mining and at a faster pace as the workload is distributed among different sites.

In recent decades, although many advancements and research have been conducted in DDM, it is still considered more challenging and complicated than traditional DM. The major concerns that arise in DDM (Gan et al., 2017; da Silva et al., 2005) are:

- **Fewer choices:** The number of choices for mining heterogeneously distributed partitions is fewer (Kalra & Lal, 2016). This makes it challenging to mine data from such sources.
- **Effective aggregation of results:** Effectively consolidating the data mining results obtained from multiple sources is a tricky task. Thus, DDM requires a robust method capable of aggregating results from distributed sites to ensure no information loss. These DDM architectures suffer inferior performance compared to centralized architecture in terms of accuracy (Dong et al., 2009). Akbar et al. (2018); Y. Yang et al. (2019) have presented different approaches for aggregating results from distributed sources without transmitting original data between sites.
- **Learning cross-terms:** In vertically distributed data, relationships often exist between the distributed sites. There is a lack of studies addressing an efficient methodology for learning the relationship between vertically distributed sites.

There are methods that can aggregate results from different heterogeneous sources or distributed sites, but they lack the mechanism to capture the interdependence that may exist between these sites (Park et al., 2001; Y. Liu et al., 2018).

- **Reduce communication:** A DDM algorithm should aim at reducing the amount of communication needed to perform the distributed computation (Turinsky & Grossman, 2000; Bhaduri & Srivastava, 2009; Gan et al., 2017).
- **Privacy issues:** There is a need to address data security issues from the perspective of distributed sites to ensure minimum data sharing among the sites (Zhao et al., 2018; Denham, Pears & Naeem, 2020a). Mining data can possibly result in the erosion of privacy, which is one of the major concerns these days. Several techniques have been developed to combat these concerns, T. Wang, Zheng, Rehmani, Yao and Huo (2019) proposed a method that allowed data analysis while protecting privacy.

Due to inefficient utilization of distributed resources, prolonged latency period, and other constraints like privacy (Zhao et al., 2018), there is a vital requirement of data mining techniques for examining and processing these distributed data sources that take into account the distributed resources (Gan et al., 2017).

2.3 Data Stream Mining

Although some complexity is involved in learning the model in the traditional data mining approach, once a model was trained from finite stationary data distribution, there was no need to update the learned model (Homayoun & Ahmadzadeh, 2016). Recent trends have led to a proliferation in generating a massive volume of data emerging as streams; the need is to analyze them in real time. A data stream involves a series of data elements ordered in time; thus, in stream settings, data instances arrive at a high pace

Table 2.1: Difference between traditional and stream processing.

Traditional Processing	Stream Processing
Offline processing.	Real-time processing.
Multiple passes of scan through the data.	Single pass scan through the entire data.
Processing time is unlimited.	Processing time is restricted.
No limits on memory	Memory usage is restricted.
Targets to achieve accurate results.	Approximate results.
Storage of data.	Data storage is infeasible.
Normal and slow data generation.	Rapid and fast data generation.

and in a consecutive manner over time, forming continuous data streams (Webb, Hyde, Cao, Nguyen & Petitjean, 2016). The instances here can be stored for a while and are mainly processed only once. The difference between traditional and stream processing has been encapsulated in Table 2.1. There has been tremendous growth in the number of applications that urge the requirement of the mining process in a dynamic setting (Yasumoto et al., 2016).

2.3.1 Processing data streams

Most learning algorithms expect the resources required for the computation to be limitless, and the entire data is available in the main memory. In the present scenario, there are tremendous applications all around us that are producing distributed data from intensifying number of smart devices (Cormode, Muthukrishnan & Zhuang, 2007). Data in these cases is flowing rapidly at a continuous rate in active and time-changing settings. The mining algorithms for such an environment must constantly process incoming data and incorporate a mechanism for detecting changes (Bifet & Gavaldà, 2007).

The learning process can occur in two ways; *offline and online* (Bessa, Miranda &

Gama, 2009). The *offline learning* process requires the whole training data to be accessible at the training time (Monteiro et al., 2009). This approach has been discussed in detail in Chapter 5. The learned model can only be used for predicting when its training has been completed. In contrast, in *online learning* (Pechenizkiy, Bakker, Žliobaitun-defined, Ivannikov & Kärkkäinen, 2010), there is no requirement for complete training data at the beginning; in fact, a model generated without complete data starts operating and is regularly updated on the emergence of more training data. *Streaming algorithms* (Street & Kim, 2001) are a class of online algorithms used to process a continuous flow of data processing examples in sequence in typically one pass. The algorithms proposed in this research work are based on online learning for streaming environments. The experimental section in Chapters 3, 4, and 6 presents a comprehensive implementation overview of this methodology. When data stream learning is made practical in a real-world scenario, the trade-off between computational resources and accuracy is inevitable. A potent mechanism should be enough to maintain a balance between the two (H. M. Gomes & Enembreck, 2014). Data streams are often accompanied by context information automatically generated by meta-data like data gathering position, datatype, and so-forth (Tekin & van der Schaar, 2013). Tekin, Canzian and van der Schaar (2014) have proposed an online adaptive learning algorithm that exploits the context vector for processing data streams, overcoming the dimensionality involved in the context information.

The volatile nature of data streams can affect the data mining models in a variety of ways (Widmer & Kubat, 1996; Abdel Wahab, 2022): *i) change in target variable (e.g., the emergence of a new class) ii) changes in the feature set iii) drift is detected*. Concept drift accounts for changes in data distribution over time; that is, the relationship between the input data and the output variable varies with time in an online supervised scenario. For more details on concept drift and its types, refer to Section 1.2. The competency of a learning system to adjust to concept drifts acts as an amplification

to *incremental learning*, which refers to a learning paradigm where learning takes place example by example (Giraud-Carrier, 2000; Alex & Nayahi, 2020). Therefore, *adaptive learning* algorithms are extended incremental learning models incorporated with an ability to modify themselves in dynamic environments (Bifet & Gavaldà, 2007). Alencar et al. (2023) proposed a novel incremental approach known as Fog-DeepStream that can efficiently process data streams in Fog computing environments and reduce data transmission between computational resources. An online adaptive learning algorithm uses the following procedure for building the model (Domingos & Hulten, 2000; Gama et al., 2014):

1. On arrival of a new instance X_t , the first step is to make the prediction, $\hat{y}_t$ using the current model.
2. On receiving the true label y_t , the loss factor is calculated as $f(\hat{y}_t, y_t)$
3. The instance (X_t, y_t) is used for updating the model.

As discussed, for streaming, it is not practical to store the data. Ikonomovska, Loskovska and Gjorgjevikj (2007) have explained different techniques for processing streams; i) *Random Sampling* allows the sampling of data stream periodically, reservoir sampling is one such method (Vitter, 1985; Gibbons & Matias, 1998) ii) *Sliding Windows* (Datar, Gionis, Indyk & Motwani, 2002) in which instead of sampling, recent data is taken into account iii) *Histograms* are used for estimating the frequency distribution of values in data stream (Poosala, Ganti & Ioannidis, 1999) iv) *Multi Resolution* uses reduction techniques like divide and conquer for the streaming environment, micro-clusters (C. C. Aggarwal, Yu, Han & Wang, 2003) is one such method v) *Sketches* are summaries that provide estimates of the frequency moments by utilizing randomized linear projections (Dobra, Garofalakis, Gehrke & Rastogi, 2002, 2004) vi) *Randomized algorithms* states using randomized methods for inspecting and sketching instead of deterministic method like probability distributions.

Some data streams may witness temporal dependencies between the class labels: in other words, when the present class label y^t is affected by preceding instance labels ($y^{t-1}, y^{t-2}, \dots$) (Gunawardana, Meek & Xu, 2011). Temporally dependent data streams account for observations dependent on each other in terms of arrival time. This effect can be realized quite often in data streams originating from smart sensor devices and data recording devices e.g., traffic sensors, environment sensors, and video surveillance (Ziffer, Bernardo, Della Valle, Cerqueira & Bifet, 2022). Bifet, Read, Žliobaitė, Pfahringer and Holmes (2013); Zliobaite, Bifet, Read, Pfahringer and Holmes (2015) have studied the predictive modeling and evaluation methods for data streams with temporal dependency. In such a learning criterion, the predictive confidence is calculated using the input feature set, including the temporal dependency factor. An empirical illustration has been attempted by Duda, Hart and Stork (2012) for predicting the maximum posterior probability in a Bayesian decision-making scenario. The best way to compute it in a time-dependent environment will be

$$\hat{y}_t = \arg \max_i P(y_t = i \mid X_t, y_{t-1} \dots y_1) \quad (2.1)$$

where $\hat{y}_t$ represents prediction at time t, i is the class. $X_1, \dots, X_{t-1}$ represents the seen observations with corresponding labels as $y_1, \dots, y_{t-1}$, considering their instant arrival. Here, the prediction at time t is dependent upon past predictions. If there is no inclusion of temporal dependence, the right side of Equation 2.1 simplifies to $\arg \max_i P(y_t = i \mid X_t)$, indicating that past predictions do not affect the prediction at time t.

For evaluating the accuracy of a data stream gradually, Bifet, Holmes, Pfahringer, Kirkby and Gavalda (2009) have discussed two methods:

1. **Holdout:** This type of evaluation is useful in cases where data has already been split into training and testing sets. It states executing the model to the autonomous

holdout test set either recurrently or using a group of examples.

2. ***Prequential or Interleaved Test-then-Train:*** Each instance is first utilized for testing the model before further being used for training. This way, testing is always done on unseen examples. The accuracy is revised gradually, eliminating the need for a holdout set for this approach. This approach has been utilized in Chapters 3, 4, and 6 for evaluating the performance of the algorithms proposed in this study.

2.3.2 Classification

In mining supervised data streams, an assumption is made that states instances from a data stream are realized as a sequence in t intervals of time units. For any instance x^t if x^t represents the vector of attribute values at time t , it is expected that true label y^t for x^t is obtained before the emergence of the next instance x^{t+1} and therefore x^t must be used immediately for training after it has been tested (M. M. Gaber, Krishnaswamy & Zaslavsky, 2004). Classification techniques have attracted the attention of a stack of researchers due to the significance of their applications (Brooks et al., 2003). In the context of streaming data, a relevant data mining problem extensively studied in literature is classification (C. C. Aggarwal et al., 2004).

A classification problem in the streaming context faces similar difficulties as any data mining model discussed above; the emergence of novel classes, concept drift, and changing of feature sets over time. Classification of news blogs would be an appropriate example of portraying these challenges. The appearance of a new topic or category in the news, like COVID-19, leads to declaring a new class. This would also induce drifts in any recommendation system that suggests news according to the user's taste because suddenly, the choice of a user may experience a change due to concept drift. Parker, Mustafa and Khan (2012) have addressed these issues using an algorithm HSMiner. The

classification problem is broken down into a hierarchy of ensemble classifiers, enabling incremental updations. Bifet et al. (2009) have proposed a framework for dealing with concept drift using two advanced methods that are variations of Bagging: ADWIN Bagging and Adaptive Size Hoeffding Tree Bagging that prunes poor performing components of an ensemble. The problem of concept drift becomes worse in the case of class imbalance. Ditzler and Polikar (2013) has proposed an ensemble-based approach for learning concept drift from imbalanced data. In one of the approaches, Learn++.Non-Stationary Environments algorithm (Karnick, Ahiskali, Muhlbaier & Polikar, 2008; Muhlbaier & Polikar, 2007) was used for monitoring concept drifts, and data imbalance was managed by using the Synthetic Minority Oversampling Technique (SMOTE) (N. Chawla, Bowyer, Hall & Kegelmeyer, 2002). Ikonomovska, Gama, Sebastião and Gjorgjevik (2009) have presented an algorithm FIRT-DD to learn the regression trees from data streams with concept drifts. A detection mechanism at each tree node is used to check for drifts based on a sequential statistical test. When a local drift is detected, the adaptation strategy gets activated by constructing a new tree and replacing it with the old one when the accuracy of the new tree becomes higher. This method allows the granular adaptation policy, thus avoiding the need to alter the global model. Class imbalance is one of the main problems witnessed in a classification problem scenario, Sudharsan, Breslin and Intizar Ali (2021) proposed a technique known as Imbal-OL, which can be used as a plugin to solve the problem of class imbalance in a range of sensor data streams.

G. Mao and Yang (2011) have proposed a classification paradigm for distributed data streams based on micro-clusters. There is an add-on to the general method of classifying distributed data streams (Masud, Gao, Khan, Han & Thuraisingham, 2008; Bifet, Gavaldà, Holmes & Pfahringer, 2018), in which the local miners divide the data chunk into an array of clusters called micro clusters and extract valuable statistics which are transmitted for global mining. After receiving micro-clusters from all the

local nodes, the global miner uses the current and last mining point information about micro-clusters to maintain step-by-step global micro-clustering.

Another work based on parallel adaptive data stream classification exploits micro clusters & nearest neighbor classification and puts forward a novel approach, "MC-NN" (Tennant et al., 2017). The application of micro clusters allows adapting to concept drifts; as such, the alteration within the clusters is performed by deleting the micro clusters that inhibit low participation. Participation is encountered by the ability of a micro cluster to absorb new data. Furthermore, work was done for parallelizing the algorithm by distributing micro-clusters to computation nodes and employing a voting mechanism (Ruta & Gabrys, 2005) for obtaining the final classification. Domingos and Hulten (2000) applies Very Fast Decision Tree (also known as *VFDT*) algorithm for mining data streams; VFDT is a Hoeffding Tree-based learning system. A novel data classification model has been introduced in (Thaw, Zhukova, Aung & Chernokulsky, 2021) for fog-enabled mobile IoT systems. These systems gather vast volumes of data from mobile devices, perform real-time processing, and deliver the data to end users based on their specific requests. This model allows to classify and excludes repeated and redundant data from raw data generated from devices, reducing the amount of data transferred to the cloud for further centralized processing. A novel technique, INFORM, was devised in (Hassani et al., 2021) to classify and enrich the IoT data streams with domain-specific metadata obtained after extracting domain ontologies of applications like air pollution.

The majority of the current work in the classification scenario has its core in one-pass mining (Domingos & Hulten, 2000; Hulten, Spencer & Domingos, 2001; Z. Zhou, Zheng, Hu, Xu & You, 2016; Sarnovsky & Kolarik, 2021). When a stable classification model is used for testing a growing data stream, in that case, the accuracy of the classification process will decrease abruptly whenever any change is noticed, like the emergence of a new class. In such cases, a classification model built on a smaller data

history would be more beneficial in terms of accuracy.

C. C. Aggarwal et al. (2004) have highlighted two problematic cases that may arise in classification. First, for classification, if the entire history of a stream is utilized, then the arrival of a new class may generate inaccurate predictions because the latest class emergence is not noticed in the rest of the stream. In this case, employing a smaller, more recent part of the stream would be more favorable. Whilst for other cases, if a new class re-emerges after an extended period, a window of long history would allow capturing the previous occurrence of that class. Hence, a robust mechanism that could aid in deciding the window size is much required (Jiao, 2006). Therefore to deal with these problems, Jiao (2006) have suggested that the incoming training data streams can be split into two parts; i) a small part of the stream is used for windowing that typically forms 1% of the data; ii) the rest of the stream is utilized for calculating the statistics related to micro clusters and class information. Although the concept of micro clustering is often used in unsupervised clustering (C. C. Aggarwal et al., 2003), but in this study, Jiao (2006) have exploited this methodology for clustering all the training data belonging to the same class.

Table 2.2 gives a summarized listing of the literary work done for classification problem scenarios.

Table 2.2: Some data stream classifiers with advantages and disadvantages.

S.No.	Algorithm	Advantages & Disadvantages
1	VFDT (Domingos & Hulten, 2000)	• Online learning from high volume data streams. • Learning in very small constant time. • Lack of mechanism to detect changes.

2	OzaBag and OzaBoost (N. Oza, 2005)	• Ability to work on online data. • Cannot handle abrupt concept drift. • Performance degrades in presence of concept drift .
3	CVFDT (Hulten et al., 2001)	• Ability to adapt to concept drifts. • Extension to VFDT • Efficiency is debased. • Doesn't consider the type of concepts. Ability to work on online data.
4	ECVFDT (Gama, Rocha & Medas, 2003)	• Improves efficiency of CVFDT. • More accurate than CVFDT.
5	Weighted classifier ensemble (H. Wang, Fan, Yu & Han, 2003)	• Works on stationary data. • Requires less time and memory. • Unable to handle concept drift and data stream.
6	On demand classifier (C. C. Aggarwal et al., 2004)	• Efficient and scalable for large data streams. • Continuous adaptation to fast-evolving data streams. • Maintain high classification accuracy.
7	ANNCAD (Law & Zaniolo, 2005)	• Fast incremental classification algorithm. • Can adapt to concept drifts. • Inability to deal with sudden concept drifts.

8	Extended Naive Bayes (Klawonn & Angelov, 2006)	• Can learn and evolve in an online manner. • Uses a number of pseudo-classes for a class. • Independence assumptions.
9	ADWIN Bagging & ASHT Bagging (Bifet et al., 2009)	• Requires less time and memory. • Unable to handle abrupt concept drift.
10	Recurrent concepts (Gama & Kosina, 2014)	• Based on single classification. • Fast adaptation after drift. • Requires context identification.
11	Incremental semi-supervised learning (Loo & Marsono, 2015)	• Quite effective and efficient. • Good at dealing with concept drift on streaming data. • Optimization method is effective and converges quickly.
12	SAND (Haque, Khan & Baron, 2016)	• Detect chunk boundaries dynamically. • Addresses concept evolution problem by detecting outliers.
13	ARF (H. M. Gomes, Barddal, Enembreck & Bifet, 2017)	• Cope with different types of concepts. • Does not require complex optimization. • Good classification performance. • Feasible amount of resources. • No improvement when all features are used to build the model.

14	VHT (Prasad & Agarwal, 2017)	• Can handle large data sizes. • Memory and computational efficient. • Small degradation in accuracy.
15	KUE (Cano & Krawczyk, 2020)	• Standard and imbalanced drifting data streams. • Have a low computational complexity. • Better predictive power.

2.3.3 Ensemble learning in data streams

Ensemble algorithms are among the most popular and easily implemented method for data stream classification (H. M. Gomes et al., 2017). When there are multiple models, and each can perform optimally on different features, to pick the best combination from all the models, ensemble modeling is required. Ensemble learning forms the core element in the execution of PODSM and MTDM (Chapters 3 and 4). In a heterogeneous distributed scenario, we have local classification results from the local model, which are built using local data on each distributed site. A distributed ensemble built as an aggregation of these local classifications using some methods like maximum confidence (discussed in Section 3.3.3) has been used as a relevant state-of-the-art approach for performance comparison with PODSM and MTDM. Ensemble learning finds its primary application in the streaming context as it can be blended with drift detection algorithms and can be adapted according to requirements like adding or removing any classifier. H. Wang et al. (2003) have described an ensemble method for dealing with concept drift problems using a weighting technique to join the various ensemble members. One requirement for ensemble learning recommended by Polikar (2006) is to use exclusive ensemble members in reference to misclassification. However, it is uncertain whether ensemble learning has a direct relation with classification performance (Kuncheva

& Whitaker, 2003). Ensemble algorithms are among the most popular and easily implemented method for data stream classification (H. M. Gomes et al., 2017).

Rijn, Holmes, Pfahringer and Vanschoren (2018) proposed a heterogeneous ensemble learning framework for data stream settings. This scheme relies on the dynamic weighting of the votes on the recent training data from the base learners. This method takes advantage of Classifier Output Difference to select the ensemble's components based on the correlation between the errors, thereby achieving diversity. An ensemble model based on Multi-Layer Perceptron (MLP), k-nearest neighbors, and RF for classification of spams and demonstrated its ability to compete with recent methods, namely Generalised Boost Regression Model, AdaBoost, and so on, was proposed by Fayaz, Khan, Rahman, Uddin and Alouffi (2020). In essence, ensemble learning techniques can be applied in a diverse range of domains such as classification, regression, and clustering (Ganaie et al., 2022). Online versions of bagging and boosting require a single pass through the training data on their arrival (N. Oza, 2005; N. C. Oza & Russell, 2001). Moreover, leveraging bagging proposed by Bifet, Holmes and Pfahringer (2010) also focuses on bagging for evolving data streams. A work by (Luong, Nguyen, Liew & Wang, 2021) is based on an observation that a subset of good base classifiers may outperform the whole ensemble structure. For predicting data under a stream environment, this method dynamically chooses an appropriate subset of base classifiers. The selection process of good base classifiers checks not only high predictive performance but also considers high confidence levels in the predictions made.

The process of learning on ensembles in a streaming context can be classified into two categories: *incremental* (Polikar, Upda, Upda & Honavar, 2001; Hulten et al., 2001; Bifet et al., 2009; Idrees, Minku, Stahl & Badii, 2020) and *window based*, where the latter is subdivided into four classes: *sliding* (Beringer & Hüllermeier, 2007; P. Zhang, Gao, Zhu & Guo, 2011), *adaptive* (Widmer & Kubat, 1996; Bifet & Gavaldà, 2007), *landmark* (Street & Kim, 2001; Kolter & Maloof, 2007), and *damped* (Silva et al.,

2013). The course of action for training an incremental learner relies on instances as they arrive, as opposed to the batch learner, which trains the learner using a batch of instances (Read, Bifet, Pfahringer & Holmes, 2012).

2.3.4 Challenges in data stream mining

Data Stream Mining is a trending subject attracting a burgeoning number of researchers. Still, there are many research holes in this arena that not only focus on space complexity but also touch on the time aspect of the streaming environment (Krempel et al., 2014; Wares et al., 2019). On top of it, a number of challenges are involved in mining data from streams, particularly in the distributed arena, which demands novel future work (Nguyen, Woon, Ng & Wan, 2012):

- Mining of huge volume of data which is continuous.
- Disclosure of change in underlying data distributions known as concept drift.
- The need for online processing of streams.

There are some prerequisites for processing in data stream settings (Ikonovska et al., 2007; Kolajo, Daramola & Adebisi, 2019):

- Continuous processing of incoming data and monitoring changes.
- There is a restriction on memory and resource usage.
- The availability of outcome in near real-time.
- Always prepared for making predictions.
- Inspecting of any record is allowed only once and cannot be held for much time.
- Only one record is processed at a time.

2.4 Heterogeneous Distributed Data Stream Mining

A mining scenario where data is vertically partitioned in a streaming environment is termed Heterogeneous Distributed Data Stream Mining. As discussed earlier, vertically distributed data signifies distributed sites having different feature sets, as shown in Figure 1.2b. The features available at various sites may be overlapping or distinct, while a common attribute usually exists that acts as an identification for each instance. Mining heterogeneous and large volume of data is the most challenging job (Villaverde et al., 2019). The most significant complication is integrating heterogeneous sources to fulfill application demands (L. Wang, 2017).

2.4.1 Sampling in heterogeneous data streams

When data sampling is applied to data streams, two main complications that arise are *i) There is no clue about the size of the data stream*; as a result, it is impossible to determine the size of the sample fraction; *ii) There is no storage of the data arriving in streams*. Consequently, a well-known technique, *reservoir sampling* (McLeod & Bellhouse, 1983; Vitter, 1985), was used to overcome these complications. It uses a randomization technique for selecting a fixed sample. The sample selected using reservoir sampling is assumed to represent the whole stream, while this assumption may not apply to streaming applications in the heterogeneous scenario, where each input stream consists of several sub-stream corresponding to different groups witnesses differing statistical properties like mean and variance. While using the traditional approach of reservoir sampling on heterogeneous data streams, there is no assurance that all sub-streams would have a statistically sufficient contribution (forms of tuples) in the reservoir; thus, the sample suffers from statistical quality issues. Also, the method is focused on maintaining one fixed-size random sample from an input stream that is unapt; for approximating each sub-stream, multiple sub-samples are required. As a

result, *stratified sampling*, which is centered on sampling from a population comprising heterogeneous groups, seems to be a good fix for the above problems (Cochran, 1977).

Above all, the heterogeneous data in the streaming context faces further complications. Firstly, it is not possible to obtain prior knowledge about the number of sub-stream with their statistical properties in a usual scenario; secondly, due to the dynamism of data streams, the statistical properties of sub-streams may alter. Therefore, a mechanism that could easily adapt to these changes is required. Al-Kateb and Lee (2014) addressed these issues by enduring a stratified reservoir sample generated on heterogeneous data streams. M. Zhang and Wang (2021) proposed a mechanism that can fuse multiple heterogeneous IoT sources in sliding windows. It uses Gaussian Mixture Model (Y. Li & Li, 2009) to assign weights to various data sources depending upon decision confidence for dynamically updating the window size and additionally implementing Multi-modality Restricted Boltzmann Machine (G. Qi, Sun, Gao, Hu & Li, 2016; Cho, Ilin & Raiko, 2011; W. Wang & Zhang, 2020) for fusing heterogeneous data streams.

2.4.2 Integration of heterogeneous data sources

A plethora of work has been done towards developing techniques for fusing these heterogeneous sources (L. Wang, 2017; Zheng, 2015). Typically, the fusion of heterogeneous sources can occur at any of the three stages; *data layer*, *feature layer*, and *decision layer*. The *data layer* is the lowest level under which the data is integrated, while *feature and decision layers* represent the middle and high-level integration (L. Zhang et al., 2018). Y. Yang et al. (2019) spotlights the knowledge graph theory used to fuse the knowledge extracted from multiple sources for power asset management problems. Another work by Yi, Zhang, Wang, Li and Zheng (2018) proposed a fusion architecture, DeepAir, that fuses heterogeneous urban data for predicting air quality. It is based on Deep Neural Networks (Miikkulainen et al., 2019), where output is generated by the weighted-merge

aggregation method.

Nawaz, Janjua and Hussain (2019) have proposed a method based on complex event processing which processes and integrates heterogeneous data streams. It is utilized as a prediction tool for cases when the underlying information is incomplete and/or uncertain using logical and probabilistic reasoning (Patkos, Plexousakis, Chibani & Amirat, 2016). In the traditional IoT architecture, data streams generated from distributed IoT devices were transferred to the remote cloud via the Internet for further analysis, raising scalability issues. The authors in (Sun & Ansari, 2016) proposed an efficient and flexible IoT architecture to collect, classify, and analyze the IoT data streams and convert them to metadata which is then transmitted to the corresponding application, providing its services to users. It intends to handle the data streams at the mobile edge.

2.4.3 Approaches for mining heterogeneously distributed streams

Decision Trees or tree-based classifiers are the most commonly used models for classification and regression (Speybroeck, 2012). There are several reasons behind their popularity; they are easy to understand, visualize, and modify. Moreover, the process of prediction is quite speedy. Finally, their ability to model the non-linear relationship classifies them as a powerful model (Chapelle & Chang, 2011; Tyree, Weinberger, Agrawal & Paykin, 2011). Their performance is excellent when they are part of ensemble learning (Fernández-Delgado, Cernadas, Barro & Amorim, 2014). A novel algorithm, Vertical Hoeffding Tree or VHT, was proposed by Kourtellis et al. (2016) to distribute the task of building a decision tree through vertical parallelism. This algorithm involved two main components; model aggregator and local statistics. The model aggregator stores the present state of the model, that is, the tree produced so far, whereas local statistics holds sufficient statistics for the attribute-value-class triplet set.

Moreover, Tennant et al. (2017) provided another approach for classifying data streams. It is based on the concept of micro-clusters and nearest neighbors following parallel execution of the stream classifier in real-time; hence can adapt to concept drift and is robust to noise.

As discussed in Section 2.2, R. Chen et al. (2001) have presented a collective approach based on Bayesian Networks for mining from heterogeneous distributed data. This was extended to the streaming context by following an online Bayesian learning approach in which the network parameters are consecutively altered using the evolving data from the streams. Bayesian Networks (Buntine, 2013; Murphy, 2002; Beretta, Castelli, Gonçalves & Ramazzotti, 2017) are a way to model relationships among features in the form of probabilities and are a good choice for learning user patterns in mining applications. The first step for online learning of Bayesian Network (Friedman & Goldszmidt, 1997) is similar to an offline approach wherein local Bayesian Networks are learned using the local data at the distributed sites, and a central site reflects the cross terms data. In order to adapt to the dynamism of streaming at each step, the local models are updated using standard models like unrestricted multinomial model (R. Chen et al., 2001), which triggers the update of the local likelihood threshold at each site. This threshold function helps in deciding on which sample should be transmitted to the central site (R. Chen et al., 2004). However, there were several limitations of this work; the structure learning process for Bayesian Network demands significant involvement; above all, this process is broader for distributed scenarios. Secondly, this mechanism assumes that the data in the form of feature sets seen at different sites are non-overlapping. Sometimes, post-processing of the overall Bayesian Network is needed to check for false local edges that may arise due to some hidden node.

Some of the literature work under the vertical distribution of data considers the heterogeneous behavior of diverse data sources (F. Wang, Hu, Zhou, Hu & Zhao, 2017; Pan, Zheng, Wilkie & Shahabi, 2013). Garg, Kim, Turaga and Prabhakaran

(2010) discusses the building up of Body Sensor Network (Hanson et al., 2009), which consists of time series data originating from various body sensors responsible for checking different types of disease symptoms. Electrocardiograms, commonly known as ECG sensors, are used to monitor heartbeat rates and look over cardiovascular-related illnesses (H. Li & Tan, 2007). These networks are used for tracking the medical condition of an individual. Each time series undergoes separate pre-processing followed by step-wise linear feature extraction, thus feeding the selections into the classification model (M. M. Gaber, Zaslavsky & Krishnaswamy, 2005).

A significant amount of work has been conducted in using web text for analyzing and predicting future trends (Varian & Choi, 2009; H. Mao, Counts & Bollen, 2011; Schumaker & Chen, 2009). The study conducted by Levenberg, Simpson, Roberts and Gottlob (2013) aimed at forecasting the trend of economic variable that is United States Nonfarm Payrolls, *a primary economic indicator* by exploiting various heterogeneous streams of web data. Every online source is treated independently as a separate data stream with its distribution. Each source is trained using features specific to each stream and forms the base classifiers. These base classifiers are then aggregated using an Independent Bayesian Classifier Combination model to obtain the global prediction (Simpson, Roberts, Psorakis & Smith, 2013; H. C. Kim & Ghahramani, 2012). This framework assumes that the various sources are independent and does not consider the cross-term effect that may exist between the various heterogeneous sources. Artificial Intelligence (AI) has revolutionized smart healthcare through recent advancements in the Internet of Medical Things (IoMT). Arya and G (2022) have proposed an ensemble federated learning (or collaborative learning). AI paradigm for classifying decentralized data streams in IoMT. This technique uses multiple clients from different edge devices or servers without exchanging raw data, thus preserving the sensitive nature of healthcare data.

Social stream entities characterize a large number of attributes; Kourtellis et al.

(2016) used VHT (Domingos & Hulten, 2000), highlighting its vertical parallelizing outlook as a good fit for social streaming mining problems. The work done by Rahnama (2014) introduced a distributed system, Sentinel, written in Java for processing and learning in a distributed manner. Moreover, it uses VFDT (Jia, 2020) based methodology for parallel classification in distributed scenarios. The authors have shown its practical implication for Twitter public stream data. Recamonde-Mendoza and Bazzan (2016) used an agent-based classification system in which the unlabelled instances were classified in batches according to the aggregation of ranks generated by the local confidence using social choice functions.

Furthermore, Salhi, Silverston, Yamazaki and Miyoshi (2019) proposed a method for early predicting gas leakage and fire risk in smart homes by collaborating data from all the devices onto a centralized location (Krishnamachari, Estrin & Wicker, 2002), following using machine learning methods (Smola & Vishwanathan, 2008) to learn the correlation between data and thereby assisting in early detection of disaster scenarios. The set of parameters included Air Temperature, Air Humidity, Liquefied Petroleum Gas (LPG), Smoke, Carbon monoxide (CO), Flame, and Carbon dioxide (CO₂). The dataset was labeled according to four risk levels; 0 for no risk, 1 for moderate risk, 2 for risk, and finally, 3 representing high risk. Sarnovsky and Kolarik (2021) proposed a heterogeneous ensemble model depending on the dynamic class weighting scheme. This model is- based on the belief that a strong model comprises a diverse collection of heterogeneous base classifiers. For a more robust ensemble, this methodology combines the dynamic weighting of ensemble members using mutual diversity score criteria. The members of the ensembles are ranked according to the diversity measures, and their weights are updated according to the diversity value; thus, the experts with higher mutual diversity are preferred. This strategy ensures ensemble modeling process robustness to evolving and changing data streams.

A two-layered architecture was proposed by (Akbar et al., 2018) for analyzing

heterogeneous IoT streams to generate an intelligent transportation system. In the first layer, the data from multiple sources were ingested and analyzed for high-level events, while the second layer considered uncertainty and proposed a probabilistic fusion of these high-level events based on the Bayesian Network. This probabilistic event processing system exploits social media data and weather to predict traffic patterns better, encouraging proactive measures. Another work (Mishra, Balan, Shibu & Hota, 2020) formulated a modular probabilistic framework for analyzing heterogeneous IoT data streams: Twitter, weather, and Traffic APIs and predicting complex event processing integration in the real-time prediction of congestion on IoT streams to address uncertainty.

The confluence of IoTs, mobile computing, cloud computing, and Big Data has propagated the intensification of streaming data in a mobile environment like smartphones, wearable devices, and others (ur Rehman et al., 2017). There are various functionalities involved in Mobile Data Stream Mining (MDSM) which are *i) acquiring data from heterogeneous data sources ii) pre-processing of data iii) fusing data iv) data mining and knowledge discovery*. Streaming data from mobile devices have a wide range of applications, such as improving business practices (Habib ur Rehman, Khan & Batool, 2016), generating customized applications for online shoppers (Tan, Lee, Wong & Ooi, 2016), and others. The massive amount of mobile data raises the question of the storage and processing abilities of the current computing devices. The most critical challenge is where and when the streaming mobile data should be processed. Different topological arrangements can be used to deploy MDSM platforms (M. Gaber, Stahl & Gomes, 2014; Abdallah, Gaber, Srinivasan & Krishnaswamy, 2015). Typically, they include various devices and systems like mobile devices, Internet and Intranet based applications, cloud data centers, and others. The authors in (ur Rehman et al., 2021) presented a device-centric streaming data analytics model for MDSM application in the mobile edge cloud computing environment. The model is built with the ability to

adaptively switch the stream execution process considering computation complexity, execution requirements, and availability of resources in far-edge devices, edge servers, and cloud computing servers.

Data stream mining algorithms that are applied to MDSM are marked as classification (Lu et al., 2012), clustering (Suarez-Tangil, Tapiador, Peris-Lopez & Pastrana, 2015), and frequent pattern mining (Rehman, Liew & Wah, 2014). MDSM applications are executed in a combination of phases, and each phase handles heterogeneity (ur Rehman et al., 2017). In the data acquisition phase, data is collected from various sources, which can be homogeneous or heterogeneous (Habib ur Rehman, Liew, Wah, Shuja & Daghighi, 2015). Heterogeneous data sources in MDSM applications refer to data streams originating from distinct data sources. Hoeffding Trees (Domingos & Hulten, 2000) is a technique used for data stream classification that generates an approximate model using Hoeffding bounds (Hoeffding, 1963) with statistically guaranteed error bounds. The increasing demand for onboard mining of data in mobile devices, which are typically executed on *resource-constrained* computing environments, puts forward the requirement of devising efficient algorithms for Pocket Data Mining or PDM. Stahl, Gaber, Liu, Bramer and Yu (2011) have proposed a collaborative approach by executing distributed Hoeffding Trees capable of processing streaming data for PDM applications. The data streams are considered to be vertically partitioned with possible overlapping features. Another similar work discussed by Kargupta et al. (2004) introduced a system known as Vehicle Data Stream Mining System, also known as *VEDAS* for continuous managing and extracting patterns from moving vehicles that generate data in the form of streams. The above-discussed methods allow personal digital assistant devices to be equipped with a mining component.

2.4.4 Challenges in HDDSM

Yasumoto et al. (2016) have discussed real-world scenarios that utilize different IoT data streams and have focused on the future demand for real-time utilization of IoT data streams instead of offline analysis. A data stream can find its place in multiple applications (Muthukrishnan, 2005); for example, remote sensing, which involves remotely monitoring the physical characteristics of land without any physical contact, is a tricky task. Such an application collects data from different sensors with varying resolutions, ultimately causing coherence in data for tasks like landslide detection and scene classification.

The most crucial challenge is aggregating and processing diverse data streams in distributed frameworks. Recently, an ongoing interest has been observed in frameworks that require distributed stream mining, on the one hand, (Mohamed, 2010) while they experience limitations on resources, on the other hand, (C. C. Aggarwal, 2007). Sensor networks and PDM, for example, restrict resources in terms of either communication or computation, or both. Data mining algorithms (particularly for clustering and classification) that can operate in a streaming environment while keeping in mind their resource utilization are need of today (Janardan & Mehta, 2017; Mena-Torres & Aguilar-Ruiz, 2014; Bandyopadhyay et al., 2006).

Currently, the data streams collected from various IoT devices are restricted to specific applications; instead, they can be applied to a variety of applications (Yasumoto et al., 2016). For instance, the weather data collected from various sensor devices, which are used for forecasting weather readings for the future, can also be used as one of the inputs for predicting traffic congestion state (Akbar et al., 2018). Even though weather data solely will not be enough for predicting traffic congestion but will have a major role in the prediction process.

2.5 Predictive Modeling via Heterogeneous Sources for Asthma

A prediction model can be defined as a statistical technique used to predict future behavior. It encompasses data mining technology that analyzes past and present data and builds a model capable of generating future outputs. These models have emerged as a powerful platform in several areas of application, like public health, clinical practice, and medical research (Steyerberg, 2009). A prediction model in healthcare plays a prominent role in facilitating medical decision-making and improving healthcare outcomes (van Smeden, Reitsma, Riley, Collins & Moons, 2021). This section gives a brief review of the relevant academic literature done in the context of asthma.

Asthma is a chronic condition that affects a significant proportion of the population worldwide and witnesses a growing trend (Bridge, Blakey & Bonnett, 2020). Its acute exacerbations or attacks are the prime cause of morbidity and mortality globally. It involves both direct costs, like hospitalization and pharmaceutical care, and indirect costs, like school absenteeism, missed work days, and productivity (Bridge et al., 2020). These exacerbations necessitate unscheduled healthcare utilization; therefore, timely and accurate diagnosis is the key to achieving excellent asthma care services.

Diagnosis of asthma attacks is challenging because it can be over or underdiagnosed. *Over diagnosis* causes costly and unrequired utilization of healthcare resources, whereas *under diagnosis* may lead to delayed treatment, morbidity, and mortality. Healthcare professionals can be assisted in decision-making by utilizing Clinical Prediction Models (CPM) (Daines et al., 2018) which adapt prediction rules to produce risk scores. CPMs use multiple predictor variables to calculate the risk for the presence of an event.

The paramount factor for asthma morbidity and mortality are exacerbations arising from asthma, and when they occur, they demand healthcare costs and services (Custovic et al., 2013; Accordini et al., 2012). A smart CPM providing appropriate services at

the right time can enhance healthcare services to patients and also helps in curtailing the cost involved in healthcare utilization. The models developed using machine learning techniques offer several advantages over traditional modeling methods (Patel, Chamberlain & Chamberlain, 2018). They can help reveal hidden interactions that a human clinician may not identify. For example, overweight/obesity was discovered as a significant factor and influential comorbidity with asthma through machine learning (Patel et al., 2018). The other merit is that the model developed using machine learning techniques is revised over time with new data.

The prediction model for asthma can be broadly divided into two types, as shown in Figure 2.2; *individual-level model* and the *population-level model*. The former class of model, also known as the characteristic-based model, makes prediction decisions for specific individuals, which also include personal data about them, whereas the latter targets a sample of the population or community as a whole. Population-based models are, however, used to make estimations on a collection of people which doesn't include characteristics of any individual. A prediction model used for predicting the risk of an asthma attack for an individual using his personal data comes into the former category of models while predicting the number of hospitalizations due to asthma falls under the population-based models. The set of predictors used as inputs under each category of models has been listed in Figure 2.2.

2.5.1 Individual based models

The ultimate goal of these models is to either analyze the parameters that trigger asthma exacerbations or build real-time forecasting for filtering out patients that possess a higher risk of hospitalizations. For early asthma treatment interference, timely diagnosis of this disease is pivotal. Asthma complications can be avoided by using proper systematized and timely treatment. The proactive work done by clinicians in determining children at

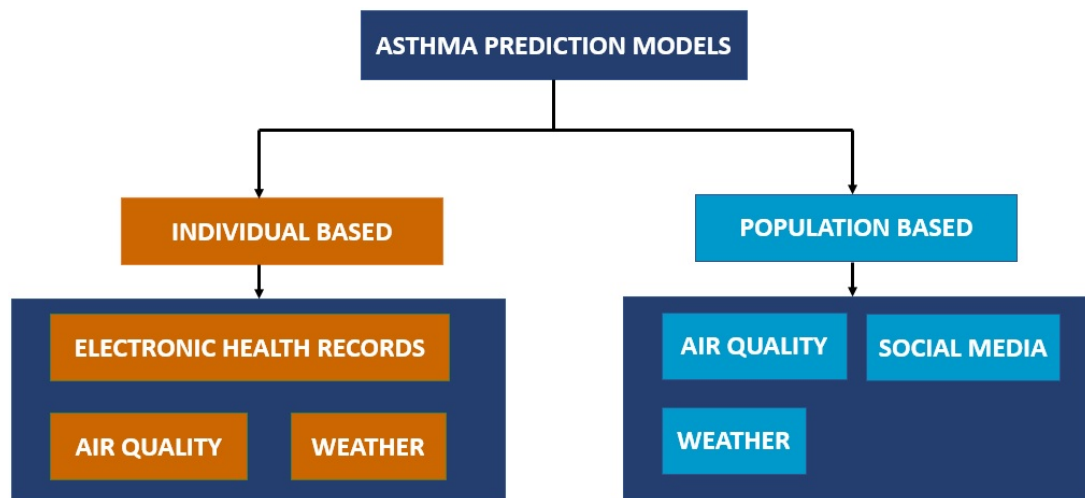


Figure 2.2: The two categories of asthma predictive modeling.

higher risk of asthma and using frequent follow-up checking would be helpful in timely pinpointing of asthma, thus initializing a treatment (Patel et al., 2018). Nevertheless, the problem is the complications faced by clinicians in making investigations among young children. Therefore, a predictive model can assist in timely asthma diagnosis, particularly in children under five years. The predictor set used for developing an individual-based prediction model can be broadly categorized as follows:

Electronic Health Record (EHR)

EHR also known as the Electronic Medical Record, is a computerized collection of a patient's health records. They comprise information like a person's age, gender, ethnicity, health history, medicines, allergies, immunization status, lab test results, hospital discharge instructions, and billing information.

A state-of-the-art study developed a model based upon more than 25000 variables for predicting the severity of asthma exacerbations in a period of 15 days (Lisspers et al., 2021). This model's long list of predictors involved comorbidity data about previous exacerbations, patient demographics, and utilized healthcare factors. For evaluating comorbidities, a measure known as Charlson Comorbidity Index (Charlson, Pompei,

Ales & MacKenzie, 1987) was used for checking the exactness score of comorbidities like congestive heart failure, Cancer, and AIDS/HIV. The authors demonstrated the potential of a detailed archive about past asthma exacerbations as a good candidate for future exacerbations. In addition to the trend of medication, frequent visits to healthcare providers due to asthma represents an event of the severe condition of a disease.

Even though several studies (Miller, Lee, Miller & Wenzel, 2007; Blakey et al., 2017) suggest age to be closely linked with future exacerbations of asthma but the work done by Divo, Martinez and Mannino (2014) reports contradictory facts assuming not age but elevated multimorbidity is the cause behind the escalated risk of exacerbations among the elderly population. For building the prediction model, four different methods were used: Gradient Boosting Trees (Extreme Gradient Boosting (XGB) and LightGBM), RF, Recurring Neural Network, and Logistic Regression. The study's primary outcome revealed that utilizing only clinical historical data may not be enough, and comorbidities also play a crucial part in predicting severe asthma conditions.

In one of the recent studies by Spyroglou, Spöck, Rigas and Paraskakis (2018), Bayesian Network Classifiers were used to forecast asthma risk using personalized patient data. Although this model was efficient in terms of accuracy, it possesses several limitations; i) the computational cost involved in developing the model becomes high when the number of parameters is large. ii) Bayesian networks are constructed assuming no dependence among variables.

Harvey and Kumar (2019) proposed an efficient system for early asthma prediction in children using children's health records plus the parents' input. The model was initially developed using 23 variables where gender, difficulty in breathing, allergies, and medication witness the highest correlation value with asthma. Various machine learning algorithms were used for building this model, with the RF classifier observing the highest accuracy of 90.9%. The need of the present time is to build a competent model to forecast whether a child is at risk of developing asthma in the future.

The prevalence of chronic disease is increasing day by day (Naghavi et al., 2017) and an expansion can also be witnessed in multimorbidity (prevalence of multiple chronic diseases) (Kone et al., 2015). Ng et al. (2020) came up with an algorithm known as CDPoRT that can predict the risk of the first chronic disease out of multiple ones while emphasizing shareable risk factors. The predictors used in this study included socio-demographic factors like age, lifestyle risk factors like alcohol and cigarette consumption, and health-related factors like body mass index. Congestive heart failure, Chronic Obstructive Pulmonary Disease (COPD), diabetes, myocardial infarction, lung cancer, and stroke were the major chronic diseases under review. CDPoRT served as a valuable tool in resource planning and decision-making by health policymakers. Another work using Western Australian Data Linkage System (WADLS) analyzed the influence of demographic factors on admissions due to chronic diseases (Brameld & Holman, 2006). Some unique predictors highlighted in this work were gender, checking if a person belongs to Aboriginal & Torres Strait Islander descent and relatively disadvantaged Census Collection District, have multiple comorbidities, locational disadvantage², and possession of private health insurance.

Although much work has been done in this direction of analyzing diverse predictor sets and predicting asthma exacerbation (Schatz, Cook, Joshua & Petitti, 2003; Hallit et al., 2019; R. Loymans et al., 2016). However, a study with a different perspective of improving the sensitivity of predictions without affecting the model's specificity has been focused by Tibble et al. (2019). The author's ideology behind this is to prevent attacks and minimize risks. The other motive was to lessen patients' worry when there is low risk involved and reduce the prescription of steroids, which have adverse long-term effects. For building the prediction model, in addition to using demographic details, practice location, the severity of asthma, smoking status, blood eosinophil count, obesity, comorbidity, previous healthcare usage, asthma control, and history of asthma

²accounts for remoteness, calculated using the road distance to specific services

attacks were the set of predictors used.

EHR plays an essential role in clinical practice, but a problem that arises when this data is used for research purposes is its inability to collect data about real-time individual socio-demographic and neighborhood factors efficiently. These factors observe a strong relationship with health (C. Chen, Weider, Konopka & Danis, 2014). Bhavsar, Gao, Phelan, Pagidipati and Goldstein (2018) augmented socioeconomic status data to EHR data with a motive to judge whether the addition of SES improves the prediction performance of health outcomes. Although they could not find many meaningful outcomes, that does not imply that socioeconomic status offers no benefit. Desikan et al. (2012) proposed the idea of supplementing EHR data with actuarial data, thus improving the precision of prediction of acute health events. Such a model has the potential to determine "less obvious" patients who are at risk of an avoidable event before the occurrence of that event. Proactive and alternative support measures can benefit such patients through recurring primary care visits and advanced care teams.

Air quality and weather data

Environment-related health issues, specifically respiratory diseases, are growing with the advancing economies. In conjunction with climatic change, air pollution also serves as a serious threat to human health. According to WHO, each year, seven million premature deaths are caused due to bad air quality and even lead to deprivation of more healthy years of life (*New WHO Global Air Quality Guidelines aim to save millions of lives from air pollution*, 2021). As asthma exhibits heterogeneous behavior where a diverse number of factors are responsible for its existence, growth, and sharpness, several factors trigger the symptoms of allergic asthma patients, like air pollution, exercise, weather conditions, and tobacco smoke (Hui et al., 2016).

Burnett, Cakmak, Brook and Krewski (1997) have highlighted the relevance of checking the effects of all the available air pollutants on health. In particular, ozone

triggers an elevation in indicators for respiratory diseases like cough, shortness of breath and others (Bascom et al., 1996). Exposure to nitrogen dioxide elevates the sensitivity to infection by a virus related to asthma (Chauhan, Krishna, Frew & Holgate, 1998). Vulnerability to oxidants and fine particles provokes the symptomatic condition leading to hospitalization (Becker & Soukup, 1999). Fusco et al. (2001) studied the correlation that may exist between daily hospital admissions for respiratory conditions and air pollutants, including SO₂, NO₂, CO, O₃ in Rome for all age groups. Data on weather attributes were also considered in this work. Poisson Regression was used to calculate the lagged effect of pollutants on asthma admission, presenting no effect for more than two days. The main finding suggested NO₂, CO, and O₃ as the leading factors contributing to respiratory hospital admissions. In essence, adults are affected by CO while NO₂ and O₃ are significant contributors to respiratory conditions in children.

A prediction model that relies solely on Peak Expiratory Flow is not entirely reliable as it misses the influence of trigger parameters like weather conditions and pollutants. D. Kim, Cho, Tamil, Song and Seo (2020) have deployed a deep learning based method, LSTM, that used Peak Expiratory Flow rates and real-time data of indoor PM₁₀ and PM_{2.5} concentration to predict the risk of asthma in children. Low-cost sensors were used to measure PM concentration at their residence to monitor PM readings. This work demonstrated the effect of indoor air quality on asthma and showed better accuracy results using deep learning models.

A methodology that would help individuals evaluate their personalized risk score for predicting the probability of an asthma attack was proposed by Do, Doig and Son (2019). If this probability were at their own threshold levels, they would receive warning alerts. With this model in place, the individual can identify their personal triggers and non-triggers. The trigger parameters included were pollen, humidity, and temperature. For example, the trigger and non-trigger list for a patient may look like this:

Triggers = [Cold<32°F, Pollen8-12, Cold<32°F + Pollen8- 12mgc, Cold<32°F +

Pollen₈₋₁₂ + Humidity<30%]

Non-Triggers = [Humidity<30%]

Authors in this work have used LSTM models to predict risk scores which helped to gain insights into personalized health.

A robust evidence base has explored varying factors and analyzed their correlation with asthma-related ED visits. A Poisson regression analysis has been applied by Babin et al. (2007) to interpret the associations between pediatric asthma-related ED visits and environmental factors in Washington, DC. These factors included daily temperature, mold count, diverse ranges of pollen, ozone, and air pollution measured in terms of PM_{2.5}. Moreover, the authors also examined the lagged effect for up to 4 days.

S. Chen and Wu (2019) have studied the effect of the environment on health, particularly considering respiratory disease. The analysis was conducted based on a stratified population by age and gender. The authors have used deep learning methods to study the impact of general air pollutants, fine particulate material, and meteorological factors on acute respiratory conditions in Beijing, China. Air pollution was found as one of the triggers that is unstable, unmanageable, and fatal (Zahran, Bailey, Damon, Garbe & Breyse, 2018). This cross-domain integration study suggested that different population groups portray their specific sensitivity to air pollution and weather parameters.

Tresa Paul, Raimond and Kanaga (2019) have proposed a framework known as Continuous Air Pollution Data Analysis, which retrieves air pollution data stored in the cloud on the fly and analyzes it using machine learning techniques. It is IoT enabled structure that continuously analyzes, monitors air pollution data, and implements a prediction model to achieve a pollution and asthma-free environment. This framework is targeted toward improving the respiratory well-being of minors. A forewarning is sent to parents, teachers, or authorities whenever it senses any sudden alteration in the atmospheric readings.

Byers, Ritchey, Vaidyanathan, Brandt and Yip (2015) analyzed the association

between different air pollutants and asthma-related ED visits in Indianapolis. This work also highlighted the comparative analysis among different seasons and age groups. The Poisson regression method was also utilized in a time series framework to learn the correlation. The parameters identified as risk factors in this study were SO₂, Ozone, PM_{2.5}, temperature, humidity, and barometric pressure. The authors identified the association of risk factors with asthma for a lag of up to 3 days. These two studies focused on analyzing the effects of various factors on asthma-related ED visits.

Using linear regression, the correlation between black carbon, a significant representative of traffic, and lung function in women was examined (Suglia, Gryparis, Schwartz & Wright, 2008). This study provided evidence of a relationship between the two and forewarned about the detrimental health implications it may have, such as poor lung function, which might lead to the development of COPD in the future. More severe symptoms plus the increased frequency of exacerbations are experienced by children with asthma who are exposed to Environment Tobacco Smoke as compared to others without exposure to tobacco smoke (Lampe, 1993). For asthma management, vulnerability to Environment Tobacco Smoke is suggested, and the best way to measure exposure to ETS is through the Biomarker Cotinine, a metabolite of nicotine. The findings of the study by Halterman et al. (2009) revealed that the level of cotinine was observed at a higher value for children whose parents or caregivers smoke when compared to the other group where parents do not smoke. In addition to this, households where there were no bans on smoking also witnessed higher cotinine levels. Particulate matter emerging from wildfire has adverse health effects, specifically respiratory illnesses like asthma exacerbations, bronchitis, and reduced lung function. Vaidyanathan, Yip and Garbe (2018) have built an online device that uses forecast information about the smoke concentration from wildfires and combines them with sensitivity measurement of the population for filtering out a population that might be at risk from smoke hazards caused due to wildfire.

Asthma patients require consistent medicinal care and management as they are affected by different triggers, like weather and air quality conditions, and the efficacy of medicines. Furthermore, work done by Ducret-Stich et al. (2013) aimed to study the impact of traffic on the air quality of nearby residential areas, which indeed causes huge effects on respiratory conditions. The authors studied the influence of the spatial and temporal pattern of traffic in the form of exhaust emissions. Different patterns of traffic were observed depending on the day of the week. This work also used meteorological data, and the outcome achieved would help in evaluating the influence of air quality due to highway traffic on health.

Ryu, Yoo, Kim and Choi (2021) developed a model for predicting unplanned re-hospitalizations within 30 days of discharge using clinical and drug prescription data about patients, weather, and air quality data of patients' residences. The models were generated using four different machine learning methods, and their performance was compared to obtain better results. These models were evaluated on external data, thus proving their generalizability. Obtaining such a prediction model with improved accuracy would assist in formulating better plans for patient care and making optimal decisions at the time of discharge.

Several studies have been conducted for predicting the hospitalizations at the time of triage (Buyuktiryaki et al., 2013; Keahey et al., 2002). These methods were solely based on clinical data, whereas Patel et al. (2018) build a model not only using general attributes but also socioeconomic status and weather-related features and neighborhood characteristics for early prediction of hospitalizations at the time of triage in pediatric asthma. The authors compared the performance of four models attributable to four different machine learning techniques, where the gradient boosting model performed the best. The ultimate goal of all of these models was real-time forecasting for filtering out patients that possess a higher risk of hospitalization on the occasion of ED triage.

Kothalawala et al. (2020) proposed the generation of two models, first with RBF

SVM classifiers using eight predictors. The other one was developed using linear SVM based on twelve predictors. Along with the direct set of predictors for asthma, some novel predictors were also added, viz age of mother at the time of child's birth, age at which solid food was introduced to the child, and total duration of breastfeeding. The addition of these novel predictors emphasizes biological factors.

A classification problem (G. Luo, He, Stone, Nkoy & Johnson, 2020) that classifies a dependent variable based on whether an asthma attack would occur or not considered 235 features, out of which 142 were finally used in the model with an aim to have better prediction accuracy over state-of-the-art approaches (R. Loymans et al., 2016; R. J. Loymans et al., 2018; Forno et al., 2010). The main characteristic predictors specific to this study comprised information regarding premature birth, allergies like environment, material or drug, and duration of asthma.

Table 2.3 exhibits the report on the state-of-the-art study conducted to understand the behavior and thus forecast adverse outcomes due to respiratory diseases, specifically asthma at the individual level. This table highlights the set of predictors used, providing information about the section of the population included in the study. The last column specifies if the work included a model for the predictions, a "✓" specifies work on prediction modeling while a "✗" indicates that the study concentrated on the analysis of data rather than the prediction modeling.

Table 2.3: Summarization of individual-level studies for asthma.

Study	Predictors used	Population focused	Predictive modeling
(Lampe, 1993)	ETS	children	✗
(Fusco et al., 2001)	Pollutants, weather data	all age-groups	✗

(Brameld & Holman, 2006)	Demographics, comorbidity, private health insurance	all age-groups	✗
(Babin et al., 2007)	Temperature, mold, pollen, ozone, air pollutants	children	✗
(Suglia et al., 2008)	Black carbon	women	✗
(Halonen et al., 2008)	Air pollutants	stratified population	✗
(Desikan et al., 2012)	EHR, actuarial data	not specified	✓
(Ducret-Stich et al., 2013)	Exhaust emission from traffic, meteorological data	children	✗
(Divo et al., 2014)	Clinical, multi-morbidity data	elderly	✓
(Byers et al., 2015)	Air pollutants, temperature, humidity, barometric pressure	stratified population	✗
(Noh et al., 2016)	Air pollutants	Stratified population	✗
(Bhavsar et al., 2018)	EHR, SES	unrestricted	✓
(Spyroglou et al., 2018)	Clinical data	all age-groups	✓
(Zahran et al., 2018)	Air pollutants, weather data	children	✗

(Vaidyanathan et al., 2018)	Smoke from wildfire	all ages	✓
(Patel et al., 2018)	Clinical, SES, weather, neighborhood data	children	✓
(Do et al., 2019)	Pollen, humidity, temperature	not specified	✓
(Harvey & Kumar, 2019)	Clinical data + parents' input	children	✓
(Tibble et al., 2019)	Health records	adults	✓
(S. Chen & Wu, 2019)	Air pollutants, meteorological factors	stratified population	✗
(Tresa Paul et al., 2019)	Air pollutants	children	✓
(Ng et al., 2020)	Demographics, lifestyle, health-related factors	adults	✓
(D. Kim et al., 2020)	PEF, weather, air pollutants data	children	✓
(Kothalawala et al., 2020)	EHR, socio-economic, biological factors of mother and child	children	✓
(G. Luo et al., 2020)	Clinical, material or drug allergies, birth information, environment, duration of asthma	stratified population	✓

(Lisspers et al., 2021)	Previous exacerbations, demographics, comorbidity data	aged > 18 years	✓
(Ryu et al., 2021)	clinical and drug prescription, weather, air quality	elderly	✓

2.5.2 Population based models

ED overcrowding is a global challenge that affects patients, as well as providers (Hoot, Zhou, Jones & Aronsky, 2007; Asheim et al., 2019). Adverse scenarios such as disease outbreaks due to influenza, for example, can cause a dramatic increase in ED visits. An unexpected rise in demand for services can lead to extended wait times and deteriorated care. A good prediction model at the population level, which is capable of accurately and timely forecasting ED visits, can play a vital role in combating overcrowding. A real-time surveillance system can alert healthcare providers and policymakers at an early stage, thus allowing them to plan resources accordingly. This would not only mitigate the operating costs of ED but also will significantly improve clinical outcomes.

Araz, Bentley and Muelleman (2014) focused on analyzing the temporal correlation between predictors for Influenza and ED visits, thus using them to develop a prediction tool. The authors discovered that data from the Google Flu trend experiences good correlation and positively impacts the prediction model's performance. Apart from trends data, the authors have used data from other sources like provider data, surveillance data, and laboratory test results data. Fan et al. (2022) utilized the Internet search index with the traditional variables using the machine learning models, aims to develop a real-time prediction model for predicting ED patient arrival.

A large volume of published studies (Whitt & Zhang, 2019; Kuo et al., 2020; Harrou, Dairi, Kadri & Sun, 2020; Harper & Mustafee, 2019) highlighted overcrowding as a

major international concern and concentrated on forecasting ED visits. Harrou et al. (2020) have demonstrated the potential of deep-learning frameworks for learning the disparity in the volume of ED visits, thereby enabling the prediction of the future trends for ED visits. The feasibility of real-time monitoring of ED visits for fostering good quality health care accessibility in the event of overcrowding was evaluated by (Hoot et al., 2007).

Another class of studies (A. J. Elliot et al., 2014; Bannister et al., 2021) analyzes the impact of thunderstorm asthma on ED attendance. In the former study, the authors reported a potential link between asthma-related attendance in ED and an occurrence of a large thunderstorm. The authors also talked about an alert system capable of predicting and alerting severe thunderstorm events, which can bring awareness and aid in planning for a sudden unexpected increase in ED attendance due to asthma. Similarly, a pilot forecasting system competent to forecast the gusty winds in severe thunderstorms and grass pollen concentrations can generate the risk of epidemic thunderstorm asthma (Bannister et al., 2021). A suite of surveillance systems (A. Elliot et al., 2014) was proposed to check the impact of heatwaves on public health, enabling early warning mechanisms and improving the health care response in the event of extreme heat.

Ram, Zhang, Williams and Pengetnze (2015) have introduced a novel method that utilizes multiple data sources to predict asthma-related ED visits for specific United States areas in real-time. The data sources used in this study were Twitter data, Google search interest, and environmental sensor data. The sensor data included air quality parameters involving six types of pollutants. The work undertaken in this paper demonstrated the application of heterogeneous data mining for health surveillance. The authors have used Decision Trees, Naïve Bayes, SVM, and ANN as the classification method to produce the prediction model and compared their accuracy. To lessen the classification errors, they have applied adaptive boosting and stacking techniques. Although no correlation was found between Google Trends and asthma-related visits in

this study, it still signifies the potential of social media data obtained from Twitter as a beneficial data source for health surveillance. In this study, the authors have emphasized anticipating ED visits for asthma in a relatively discrete geographic area. The main limitation of this work was the short time frame of study consisting of 74 days which lacked examining the effects of seasonal variations. Secondly, this work missed the concept of analyzing the lagged effect that may exist between changes in the Air Quality Index and the number of asthma-related emergency visits.

A more recent approach focused on various weather and pollutant data in Dallas County of Texas to develop a forecasting tool for respiratory diseases, including COPD, asthma, and other respiratory conditions (Khatri & Tamil, 2017). This work's study period was three years, from 01/01/2011 to 31/12/2013, which included eight parameters covering weather and pollutant measurements. Authors have used ANN classifiers and demonstrated good accuracy could be achieved using these measurements as the input variables. More insight provided details about the lagged effect of predictors on asthma-related visits and admissions. However, the study was confined to a limited number of predictors. The authors also mentioned ANN's capability to achieve correct prediction even after learning from data with noise; thus, the study didn't include any data cleaning technique.

Models that are apt enough to determine the proportion of the population at moderate or low risk of asthma attacks provide valuable insights into preventing asthma. Besides being cost-effective, they also help curtail the wastage of healthcare resources, save from adverse effects of unneeded asthma medications, and encourage self-asthma care. Table 2.4 provides a summary of closely linked existing work.

Table 2.4: Existing similar work with an aim to develop a population-based prediction model for asthma.

Author	Location of study	Disease	Duration	Acc (%)
(Ram et al., 2015)	Dallas, USA	Asthma	74 days	70
(Khatrri & Tamil, 2017)	Dallas County, USA	All resp. dis.	3 years	81

2.6 Background Work

Ensemble learning is a common approach to mining vertically distributed data and is marked as a two-stage process. In the first stage, local models are built using the local views of data which are later aggregated in the subsequent stage. There have been several ensemble learning techniques proposed by Mokeddem and Belbachir (2009); Devi (2014); McConnell and Skillicorn (2004); N. V. Chawla, Hall, Bowyer and Kegelmeyer (2004), some of which can be applied in streaming context (van Rijn, Holmes, Pfahringer & Vanschoren, 2015; Museba et al., 2021; Sarnovsky & Kolarik, 2021), but not all of them can work for streaming data (Tumer & Ghosh, 2000). The most common method to merge the classification of local classifiers is the maximum confidence aggregation method (L. Li, Zou, Hu, Wu & Yu, 2013), which picks the base learner with maximum confidence as the final prediction result. Tumer and Ghosh (2000) used a method based on order statistics for merging the results from various distributed classifiers. The combiner obtained from the order statistics combination can outperform simple combiners when a sufficient amount of variability is seen in the classifiers. The potential of an ensemble lies in the diversity of its component learners (Kuncheva & Whitaker, 2003; Idrees et al., 2020). Two classifiers are considered diverse if their error on the new data points differs. The main difficulties that arise while framing an ensemble learner are the choice of learning algorithm used to train the base learners to achieve maximum diversity (Minku, White & Yao, 2010) and second is what method should be used to fuse the different results from the individual base learners

(Beyer, Heesche, Hauptmann, Otte & Kruse, 2009; L. Zhang et al., 2018). This class of meta-learners is a robust method as they are less affected by outliers with no botheration of tuning additional parameters in meta-learners. Besides this, Basak and Kothari (2004) proposed a method for combining the outputs from component classifiers, in which the final posterior probability is calculated using the posterior probability predicted by the individual classifiers.

A basic decentralized structure that is distributed ensemble for a classification problem (Canzian, Zhang & van der Schaar, 2013) suggests that each distributed site learns a local model, producing a local classification. These classifications are combined using different aggregation methods like maximum confidence, majority voting, and stacking (Rokach, 2010). Distributed ensemble, even though does not require any data communication, lacks a mechanism for learning cross-terms that directly effects the accuracy of the classification process in a heterogeneous distributed environment.

Park et al. (2001) proposed a method to mine data effectively in a heterogeneous environment while distributing the computational load and reducing the communication cost when compared to the centralized processing architecture. Each local site (distributed node) builds its local model using a decision tree, and the global model is constructed at the global site. Instead of centralizing the whole data, the central global tree is constructed based on a subset of records transmitted by the local sites. Now the question arises, “*Which records would be transmitted to the central site?*”, for this the authors used setting a value called ***confidence threshold***. Thus, the records classified with a confidence lower than this value are termed *trouble records or unconfident records* and are transmitted to the central (global) trouble site. When a distributed local site declares a record as trouble, it means that this site is unable to confidently classify this record and send its record fragment to the trouble site for processing. A global trouble site refers to a site that aids in classifying these trouble records. This global site processes the records only if it receives the record fragment from all the local sites.

The classifier generated at this site captures the relationship (referred to as *cross-terms*) that exists within the data present at the distributed sites, i.e. patterns that can only be learned by viewing the data from the collective perspective of more than one site. Thus, this architecture provides a way to maintain a balance between *transmission cost* and *accuracy*. The rationale for selecting the term "trouble" in this context is derived from its usage in the state-of-the-art foundational research in (Denham, 2019; Denham, Pears & Naeem, 2020b). In (Park et al., 2001), trouble records are denoted by "bad" rows.

However, there were some limitations in the methodology proposed by Park et al. (2001); the centralized site processes only those records declared unconfident by all the sites. This allows the centralized site to capture the cross-terms or relationships between all the sites. Therefore, the transmission of other records for which only a subset of distributed sites are unconfident are not processed by the centralized site. These records serve no purpose and result in uneconomic communication.

Subsequently, Denham et al. (2020b) came up with an architecture named Hierarchical Distributed Stream Miner (HDSM) using Parks' method (Park et al., 2001) as its foundation. HDSM has its basis in capturing the cross-terms between the subset of sites. Each distributed node or primary site constructs a local classifier based on its data view and identifies trouble records (records that are classified with a confidence value less than the threshold). The primary site corresponds to the distributed sites, each originating from different data stream (heterogeneous distributed data streams). Each primary site contains a different set of features (or attributes); however, there could be overlapping between features available at different sites. Instead of a single centralized trouble site, it builds a hierarchy of trouble sites, where each trouble site is formed by pairing two source sites. This method allows the creation of multiple trouble sites, each receiving trouble records from a different subset of source sites (a source site can be with either a primary site or another trouble site). Every trouble site is of certain order depending upon its level on the hierarchy. As such primary sites are of order 0, and the

Table 2.5: Comparison among the state-of-the-art foundational work.

Methods	Advantages	Disadvantages
Distributed ensemble	• Least communication	• No mechanism to learn cross-terms • Less accurate
Park's Method	• Mechanism to learn cross-terms between all sites • More accurate than Distributed ensembles	• Wastage of communication • Usage of a single trouble site for all distributed sites
HDSM Method	• Can learn cross-terms between subsets of sites • Less communication wastage than Parks' method • Better trade-off between accuracy and communication	• Still wastage of communication • Brute force strategy used in learning cross-terms

order keeps increasing on moving down the hierarchy. The system is initialized with a minimal structure consisting of only the primary sites and the aggregator with no trouble sites. The aggregator site in this architecture receives the classification results from all the sites and generates the final output. A trouble site is created when the agreement between two sites becomes greater than the threshold value. *Agreement* measures the likelihood when both the source sites agree that any received record is trouble (refer (Denham et al., 2020b) for more details). The architecture is equipped with three sets of monitors, namely the creation, removal, and blacklist monitors, responsible for creating, removing, or blacklisting a trouble site respectively dynamically depending upon the requirements.

Table 2.5 presents the comparative study among the state-of-the-art foundational work, which clearly indicates the pros and cons of each study. This foundational work will be used for comparing the performance of PODSM and MTDM in Chapters 3, 4, and 6.

2.7 Research Gaps

Our literature review focused on distributed stream mining with an emphasis on heterogeneously distributed data; a part of the chapter highlighted the work done towards prediction modeling for asthma. The research work reviewed was based on specific domains listed under Section 2.1. The first three domains reviewed work done in DDM in an offline context, data stream mining, and HDDSM, which helped to examine a considerable amount of literature in this arena. Under the fourth domain, many studies concentrating on predictive analytics for asthma were compared and the findings were summarized in Tables 2.3 and 2.4. These tables provided details on the type of predictors and modeling used. The background work discussed in Section 2.6 evaluated the closely aligned work that forms the foundation of this study. The outcome of this evaluation was encapsulated in Table 2.5, highlighting the pros and cons of each. After going through the rich state-of-the-art studies, we have identified a set of research gaps in the space of mining heterogeneously distributed data streams and asthma that have been outlined below:

1. **Wastage of resources** In a heterogeneous scenario, an effort to shift to distributed architecture for processing the heterogeneous distributed data would cause a detrimental effect on accuracy. Thus, the main requirement for an algorithm designed for distributed architecture is to maintain *a balance between accuracy and resources*. A centralized architecture (R. Chen et al., 2001; Salhi et al., 2019) requires the fusion of data from all the distributed sites. Even though this architecture involves complete data centralization, thus introducing massive utilization of resources but on the other hand, the accuracy achieved by having such an architecture in place is high. Therefore, efforts were made by Park et al. (2001), R. Chen et al. (2004), and Denham (2019) to decrease the usage of resources, in other words, for curtailing the data transmission from the distributed

sites. Although these studies have decreased data transmission, a major portion of the transmission is still wasted and has no usage. Research contribution to reducing wastage and decreasing communication and computational cost has been made in Chapter 3.

2. Learning cross-terms The primary cause that affects the accuracy in a decentralized architecture is the lack of a mechanism to learn the cross-terms between the heterogeneous distributed data sites. There is some direct or indirect relationship between the heterogeneous data present at distributed sites which can only be captured when there is a complete copy of data from all the sources/sites at a single site. This is the most crucial problem witnessed while mining data from distributed sources; any attempt to learn the cross-terms leads to the utilization of resources. The key concept here requires the integration of heterogeneously distributed sources. Integration at the data level (Salhi et al., 2019) involves massive resources, while an integration at the decision level (L. Li et al., 2013) greatly impacts accuracy. Therefore, the need is to come up with an intermediate solution (Park et al., 2001; R. Chen et al., 2001; Denham, 2019) that can learn the cross-terms while preserving the balance between the *resources utilized and accuracy*. The existing studies dedicated to learning the cross-terms without involving complete centralization encounter different drawbacks, as discussed in Section 2.6. The contribution made in addressing this research gap has been discussed in Chapter 4.

3. Conducting predictive analysis for asthma in NZ. The prediction model for asthma can be split into *individual and population* models. As discussed, an *individual-level* model generates predictions for individuals, while a *population one* produces the prediction results at the population level. Several studies have examined an array of parameters to study their correlation with asthma-related

hospital data. As a result, the highly correlated parameters are used as input for the prediction model. The location of the study, period, and the list of parameters used as input have a major impact on the model's performance. As such, the correlation of the parameters with the hospital data differs according to the geographic area under study. There is a dearth of studies analyzing the correlation of multiple heterogeneous parameters with the hospital data related to asthma, henceforth using them to develop a prediction model at the population level for NZ. In fact, NZ is one of the countries with high rates of asthma in the world. Moreover, asthma is a heterogeneous disease affected by different factors such as environment, genetics, and clinical. So, developing a prediction model for asthma requires data from heterogeneous sources which are geographically distributed to obtain satisfactory performance of the model. The existing studies miss a few parameters that may affect asthma conditions. For predictive modeling, less work has been done towards pre-processing data and balancing the imbalanced data. As per our knowledge, no study has been concentrating on devising the population-based asthma model in NZ. These gaps have been addressed in Chapter 5.

4. **Verifying decentralized algorithms** on a live dataset built for asthma case. The current prediction model developed for asthma uses centralized architecture for mining the heterogeneously distributed sources (Ram et al., 2015; Khatri & Tamil, 2017). Thus, it involves huge utilization of resources in the mining process. There is an absence of work towards replacing a centralized approach with a decentralized structure and checking what difference it creates by implementing decentralized algorithms. Overcrowding ED is a prevalent problem worldwide. This problem requires adaption to change in demand in real-time while continuing to treat patients. Places like Auckland, NZ, have uncertain and regularly changing weather conditions, ultimately causing uncertainty in the number of

admissions or ED visits due to asthma. Previous studies have looked into the offline processing of heterogeneous sources. The overall process of applying decentralized architecture in an online context and testing the applicability of the proposed algorithms have been targeted in Chapter 6.

Figure 2.3 displays the panoramic view of the thesis indicating the research gaps that have been identified through an extensive review of the literature with the associative contributions made in that direction. As discussed above, the literature review was framed into two parts: HDDSM and an asthma case study. This figure illuminates both parts, clearly formulating the solutions with their benefits and addressing the specific research gap.

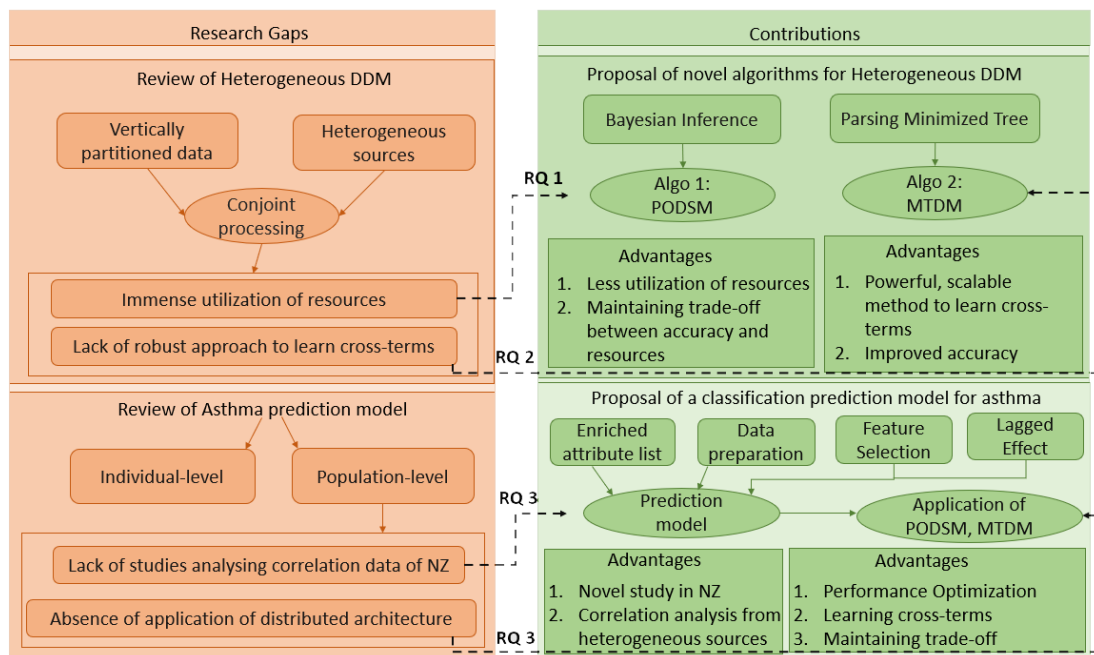


Figure 2.3: Diagrammatic illustration of mapping from research gaps to research questions to research contributions.

2.8 Chapter Summary

In this chapter, we highlighted the literary works undertaken under data mining, strictly speaking in a heterogeneous distributed data stream context. The chapter was distinguished into two parts; where the first part discussed the related literature work on DDM, data stream mining, and HDDSM. In the second phase, we reviewed the profound background on asthma and the models developed to provide proactive warnings on adverse effects like signaling an asthma attack or hospital overcrowding due to asthma exacerbations. The foundational work that forms the basis for this research work has been well communicated in this chapter. Furthermore, the shortcomings in the state-of-the-art methods were identified and pinpointed, which helped in identifying the research gaps.

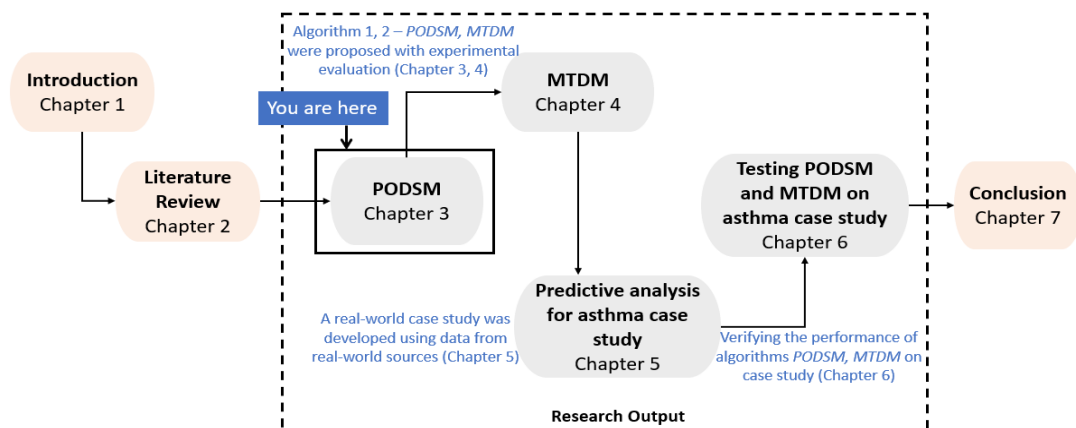
This chapter provided in-depth documentation that can assist in developing a clear understanding of the latest work done in this area. It reveals that though there has been extensive research conducted in DDM, the specific area of mining vertically partitioned data still lacks appropriate algorithms for processing the heterogeneous distributed data streams.

On the other hand, many prediction models have been developed at the individual level for asthma that requires identification of each individual under study and provides warning related to an individual. Nevertheless, at the population/community level, these prediction models require more researchers' attention. Also, the geographic area offers different research opportunities in health forecasting models.

Chapter 3

Performance Optimizer in Distributed Stream Mining (PODSM)

This chapter presents the comprehensive details of the PODSM algorithm in the form of a theoretical explanation with implementation. It aims to optimize performance in mining heterogeneously distributed data streams, addressing Research Gap 1. The chapter concentrates on decreasing resources involved in the mining process. A comprehensive discussion on PODSM methodology has been presented, along with a thorough experimental evaluation on eleven datasets and a comparative analysis with closely related work.



Roadmap of the thesis.

3.1 Introduction

Several attempts toward DDM have focused on addressing the heterogeneous nature of data sources; however, such approaches do not prioritize the reduction of data communication costs which could be prohibitive in large-scale sensor networks. The key aspect in heterogeneously distributed environments is *high communication and computational costs*. Previous researchers, Park et al. (2001); R. Chen et al. (2001); Recamonde-Mendoza and Bazzan (2016); Y. Liu et al. (2018); Denham et al. (2020b) have established that an attempt to decrease communication load in such an environment has a negative impact on classification accuracy. Therefore, maintaining a balance between *resources utilized* and *accuracy* is the pivotal concern while mining data from the indicated scenario. This chapter proposes an algorithm, *Performance Optimizer in Distributed Stream Mining (PODSM)*, based on *Bayesian Inference* to reduce the communication volume and resource time in the HDDSM environment while retaining prediction accuracy.

3.2 Problem Definition

In vertically or heterogeneously distributed data, each site essentially has a different view of the same set of records and shares at least one “unique key” that can be used to join vertical fragments of records from different sites. It is worth mentioning here that the distributed sites are unaware of each other’s state; expressly, a distributed site knows only its view while it has no information about peer sites’ feature set, their values, and the confidence configuration. Work by Park et al. (2001) and Denham et al. (2020b) proposes the creation of trouble site/trouble sites for processing trouble records. Trouble records are bad records that are classified with low confidence by any site, and trouble sites are the sites that are generated to confidently classify these trouble records. As

such, these trouble sites are generated by combining features of two different distributed sites (refer Section 2.6 for more details). In *Parks' method*, (*Park et al., 2001*)¹, a single trouble site is created which processes only records that are declared *trouble* by all the primary sites; putting it in another way, the transmission of all records for which any of the distributed sites is confident have no value at the trouble site. On the other hand, in *HDSM* (*Denham et al., 2020b*)², a trouble site is formed by combining two source sites and processes a record only if it receives the record from both of its source sites on being declared as a *trouble record* by both of them. Thus, implies that the training set of a trouble site is the collection of the merged record containing all the features of its two source sites.

While HDSM has reduced the unnecessary communication at the trouble site by capturing *cross-terms* only between subsets of distributed sites, unlike Parks' approach that attempts to capture cross-terms between all sites. But, there still remains a substantial communication cost that is not utilized or serves no purpose. This can be demonstrated by an example. Suppose *A*, *B*, and *C* are three distributed sites. There can be eight different scenarios reflecting the confidence status of these sites, as depicted in Table 3.1. In Parks' method, sites *A*, *B*, and *C* would be connected to a single trouble site; let us label it as T_{ABC} as shown in Figure 3.1a. The site T_{ABC} would receive trouble records from all three sites. Therefore, T_{ABC} will only process records that have been declared trouble by all the sites, as represented in the first case in Table 3.1. For all the other cases, the transmitted records are valueless except for the case when all the sites are confident, as there will be no transmission in that case.

On the other hand, in HDSM, rather than a single trouble site connecting all the distributed sites, there are multiple trouble sites combining sites in pairs of two. Let us suppose T_{AB} is the trouble site generated from sites *A* and *B* capturing the cross-terms

¹This work would be referred to as Parks' method/approach for rest of the thesis.

²HDSM will be used to refer (*Denham et al., 2020b*) work.

Table 3.1: Summarization of transmission and processing status at trouble site for different possibilities of confidence status for three sites; A, B, and C.

Site Status			Trans. Status			Proc. Status	
Site A	Site B	Site C	A	B	C	Parks'	HDSM
U	U	U	✓	✓	✓	✓	✓
C	C	C	✗	✗	✗	✗	✗
U	U	C	✓	✓	✗	✗	✓
U	C	U	✓	✗	✓	✗	✗
C	U	U	✗	✓	✓	✗	✗
U	C	C	✓	✗	✗	✗	✗
C	U	C	✗	✓	✗	✗	✗
C	C	U	✗	✗	✓	✗	✗

between A and B, while $T_{[AB]C}$ is another trouble site generated from T_{AB} and C, hence capturing cross-terms between them as shown in Figure 3.1b. The state of transmission for all the cases has been summarized in the same table, Table 3.1. When both the paired sites are confident, there will be no transmission to the trouble site, but when one is unconfident³, then only the unconfident site will send its record fragment to the trouble site. As a result, the trouble site holds record fragments from only one site, which alone will not be processed at the trouble site. This results in unnecessary overhead of communication, thereby decreasing productivity. When both the sites are unconfident, the trouble site receives the record fragment from both of them and processes the merged record. It can be seen that with HDSM (Denham et al., 2020b), there have been some savings in communication, but there still exists ample scope for improvement in communication cost. The last two columns in Table 3.1 depicts the processing status at the trouble site. According to the explanation given above, HDSM has saved communication costs for one case when A and B are unconfident while C is confident. T_{AB} would be processing this set of records and forwarding its result to the aggregator. Except for the first, second, and third rows, it can be seen that there

³This term is used when the classification confidence level of a site is less than the threshold for a record.

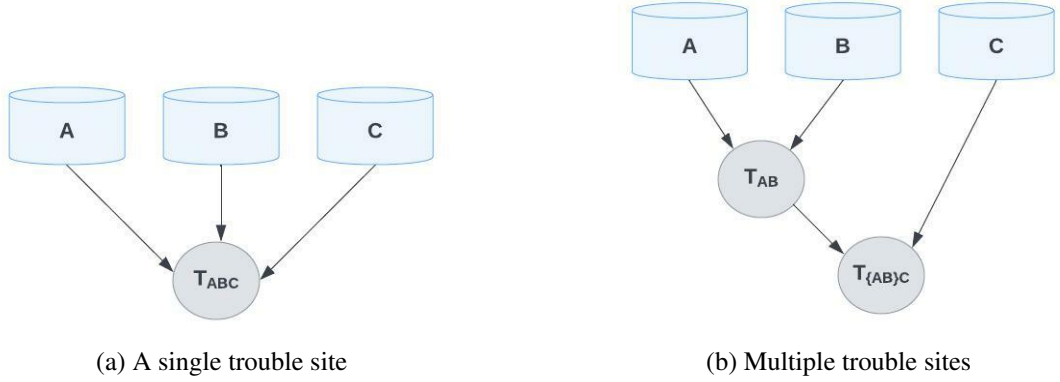


Figure 3.1: Architecture used for processing distributed sources using decentralized algorithms; a) a single trouble site is used for capturing cross-terms between all the distributed sites, whereas in b) a hierarchy of trouble sites is generated in streaming context for capturing cross-terms between a pair of sites.

are transmissions of records but no corresponding processing at the trouble site, which shows *wastage of resources*.

3.3 Architecture

The following section elucidates the architecture of the PODSM algorithm for classification in a vertically distributed data stream scenario. It aims at curtailing the communication between the sites and enhancing the efficiency of computational resource time while maintaining the accuracy level. PODSM has its foundation in HDSM architecture reviewed in Section 2.6. Figure 3.2 depicts the architecture used in PODSM for processing heterogeneous sources or vertically distributed sites. It consists of several primary sites, trouble sites, and an aggregator site. Every primary site receives input from heterogeneous data streams that differ in feature sets. It is assumed that each feature set will contain a common key feature not used for classification but allows record fragments and classifications to be merged at trouble sites and aggregator, respectively.

The ultimate motive of refining productivity in communication and computation has been addressed by targeting the records sent by only one unconfident source site.

These records have no significance in generating the classifier at the trouble site and result in wastage of communication. The basic ideology behind this algorithm is that if a source site is unconfident for a record, it predicts the probability of paired source site being confident. If the predicted probability value is greater than the threshold, the unconfident site would assume that the other site is confident and therefore makes a decision not to send the record to the trouble site. While in the other case, when the prediction value is less than the *threshold* the unconfident site would send the record. Suppose, A and B are the two paired source sites, and T_{AB} is the trouble site formed after merging these two sites. For a particular record r , A & B's views for this record are recorded as r_A, r_B respectively. If A is unconfident for this particular record, then it would predict the probability of B being confident. However, the major issue here is that site A knows about its view, r_A , whereas r_B is out of scope for this site.

For calculating the probability of other paired site being confident, PODSM exploits the *Bayes' theorem*, according to which the conditional probability can be calculated as:

$$P(A | B) = \frac{P(B | A)P(A)}{P(B | A)P(A) + P(B | A^c)P(A^c)} \quad (3.1)$$

Bayes' theorem is the fundamental way to calculate conditional probability. *Conditional probability* calculates an event's probability, given that another event has already occurred. It is denoted by $P(A | B)$ and expresses *the probability of A given B*. Therefore, the left side of Equation 3.1 gives the conditional probability of A given B. In the denominator of the right side in the same equation, A^c corresponds to the complement of A, expressly it is $P(not A)$ or $1 - P(A)$. This denominator term is based on the expression $P(B) = P(B | A)P(A) + P(B | A^c)P(A^c)$, which states that the probability of event B is the sum of the conditional probabilities of event B given that event A has or has not occurred. The simplified version of Bayes' theorem is denoted in Equation 3.2.

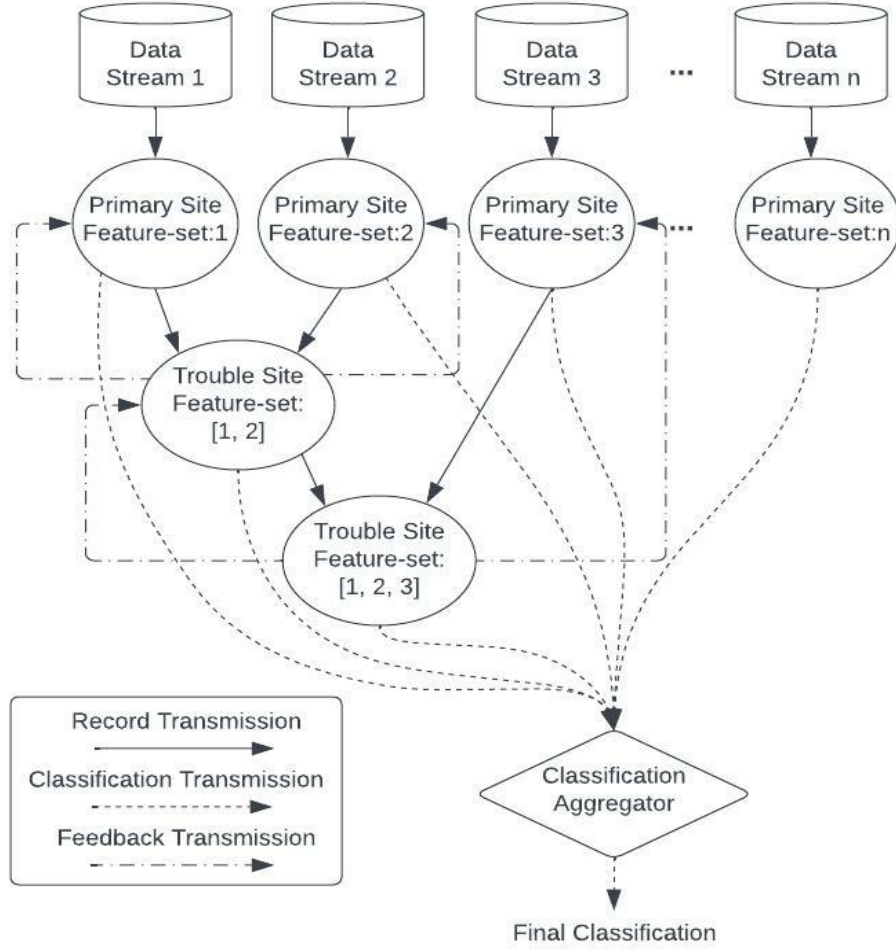


Figure 3.2: Pictorial representation of the architecture used in PODSM, consisting of primary sites, trouble sites, and the aggregator.

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)} \quad (3.2)$$

In general terms, the Equation 3.2 can also be rephrased as:

$$Posterior = \frac{Likelihood * Prior}{Evidence} \quad (3.3)$$

More details on Bayes' theorem can be referred to Joyce (2021).

We will use Equation 3.1 for calculating conditional probabilities in this algorithm. For our case, we have two unlike, blind source sites with different views and different

local models, which is unshareable among sites. Using this Bayesian Inference, we are trying to estimate the probability of a paired site being confident by an unconfident site. To exemplify, if we have two source sites, A and B, with feature sets X and Y , respectively, as shown in Figure 3.3 and T_{AB} is the trouble site. Suppose for a particular record, site A is unconfident. Before sending its record fragment to T_{AB} , it would predict the probability of B being confident, given that A is unconfident. The formula that would be used by A to estimate the probability of confidence of B is:

$$P(B \text{ conf} \mid X = x \text{ } A \sim \text{conf}) = \frac{P(X=x \text{ } A \sim \text{conf} \mid B \text{ conf}) P(B \text{ conf})}{P(X=x \text{ } A \sim \text{conf} \mid B \text{ conf}) P(B \text{ conf}) + P(X=x \text{ } A \sim \text{conf} \mid B \sim \text{conf}) P(B \sim \text{conf})} \quad (3.4)$$

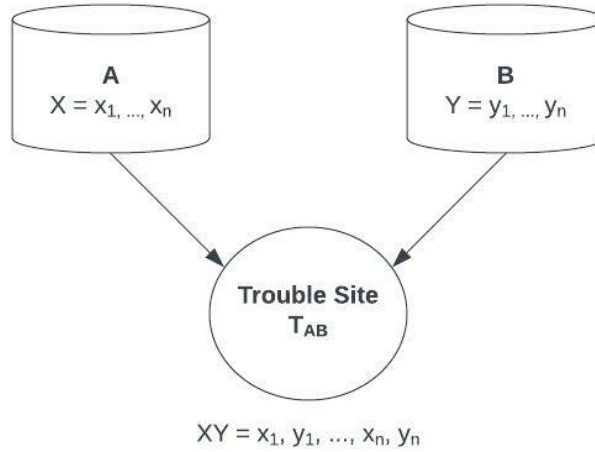


Figure 3.3: Structural illustration of sites for calculating statistics; it consists of two primary sites, A and B, with feature sets represented by X and Y , respectively. Trouble site T_{AB} is formed by merging features from A and B.

The terms on the left-hand side will give the probability of B being confident when A is unconfident for $X = x$, where X is the set of attributes at site A and x contribute to values of attributes for that particular record, while the terms on the right side are

calculated at the trouble site using the past data (i.e., a batch of seen records). Let us discuss it in detail; when the implementation commences, a batch of records is allowed to flow according to HDSM without any restrictions, as summarized in Table 3.1. That is, the unconfident site continues to send its record fragment to the trouble site irrespective of whether their paired source site is confident or unconfident, thus inducing unfruitful communication. At the end of this batch of records, the trouble site calculates statistics for the terms mentioned on the right side of the above Equation 3.4 and sends these statistics to the source sites. Thus, the trouble sites are responsible for calculating and transmitting these statistics to their respective source sites. The source sites use statistics or feedback from the trouble sites when they encounter a trouble record, strictly speaking, a record for which these sites are unconfident. As a consequence, a feedback system is incorporated into the sites, enabling them to make predictions about the confidence status of their peer sites. This feedback serves as a foundation for enhancing their performance.

Likewise, when B is unconfident for a record, it uses a modified version of the formula mentioned in Equation 3.4 to predict the probability of A being confident, but in that event, positions of A and B get interchanged, Equation 3.5 represents the modified version of Equation 3.4.

$$P(A \text{ conf} \mid Y = y \ B \sim \text{conf}) = \frac{P(Y=y \ B \sim \text{conf} \mid A \text{ conf}) P(A \text{ conf})}{P(Y=y \ B \sim \text{conf} \mid A \text{ conf}) P(A \text{ conf}) + P(Y=y \ B \sim \text{conf} \mid A \sim \text{conf}) P(A \sim \text{conf})} \quad (3.5)$$

We will use an example to demonstrate the working of PODSM; let us suppose a distributed data scenario with 20,000 records and the same site structure as shown in Figure 3.3, A and B represent the sites with feature sets X and Y, respectively. For

this case, we will set the size of *batch-1*⁴ as 10,000 records. Till 10,000 records, sites A and B will send their trouble records to T_{AB} regardless of the confidence status of each other. This set of records will be used by the trouble site i.e. T_{AB} , to calculate the values specified on the right side of Equations 3.4 and 3.5. At the end of 10,000 records, these values will be transmitted by T_{AB} to A to B, respectively. From the 10,001st record, when sites A and B will encounter an unconfident or trouble record, they will calculate the left side of Equations 3.4 and 3.5 to predict the confidence status of each other and make the decision about whether to send their own trouble record or not. The statistics or feedback values are calculated from a specific category of records; for example, $P(X = x \mid A \sim conf \mid B \sim conf)$ is calculated from records for which A is unconfident, and B is confident whereas $P(B \sim conf)$ and $P(B \sim conf)$ accounts for B's total confident and unconfident records respectively, independent of A's confidence status. Lastly, $P(X = x \mid A \sim conf \mid B \sim conf)$ correspond to records for which both A and B are unconfident. The conditional probabilities include an expression " $X = x$ "; these represent the record set of site A, which have been transmitted to trouble site T_{AB} when B is confident, as represented in $(P(X = x \mid A \sim conf \mid B \sim conf))$ or B is unconfident in $(P(X = x \mid A \sim conf \mid B \sim conf))$. In both of these cases, T_{AB} would have access to record fragment of A ($X = x_1, \dots, x_n$), especially under the first batch. There must be ample records for calculating any of these statistics to avoid obtaining an inappropriate value, such as 0. Substituting these statistics on the right side of Equation 3.4 would assist any unconfident site in predicting the confidence status of its peer site. The transmission of statistics values from trouble sites to the source sites for decision-making purposes has a relatively minor impact compared to the transmission of actual records. As a result, it has minimal effects on the overall transmission time.

For achieving dynamism for the data stream environment, the statistics calculated at

⁴First set of records used by the trouble site as the historic or past data for calculation of the conditional probabilities.

the end of the first batch are regularly updated on the incoming of every record. This makes the algorithm adaptive for situations when there is any change in the structure, expressly if there is any addition or removal of trouble site, and further adds flexibility on account of concept drifts in data streams. After the first batch, the algorithm regularly checks the state of the network; if any trouble site has been created, it runs the required procedure to calculate new statistic values. Otherwise, it only updates the existing statistics.

The decision made by the unconfident site depending upon the calculated probability value, is assumed to be accurate; however, there may be occasions when this decision may turn inaccurate. In the case of a correct decision, communication is saved, but if the decision is not correct, even though it would reduce communication, but may also produce a detrimental effect on accuracy. *For instance*, if A and B were unconfident for a record and A incorrectly predicted B to be confident. In this case, A would skip transmission of its record to the trouble site, making a wrong consideration of B to be confident. According to protocols, when both sites are unconfident, they must send their respective record fragment to the trouble site for better classification results. In this case, the trouble site won't receive the record from site A, so the record will not be processed at the trouble site, thereby may affect accuracy. The following section gives a detailed explanation of the algorithm for the above-discussed method.

3.3.1 PODSM algorithm

The PODSM algorithm begins with primary sites generating their local classifier and correspondingly having their local *confidence threshold* value. When a new unlabelled record arrives, it calls a procedure `processRecord` in Algorithm 1. Suppose the local classification confidence of the record is less than the confidence threshold, and it belongs to *batch-1*. In that case, that particular record is forwarded to the trouble

site for calculating the statistics. On the other hand, if the record does not belong to *batch-1* and there exists sufficient statistics (received from the trouble site as explained above), then the unconfident local site uses statistics or feedback data from the trouble site to estimate the probability of the other site being confident. If the calculated value > 0.5 , the unconfident site makes the decision not to send the trouble record to T_{AB} . In the other case, the unconfident site will send its trouble record to T_{AB} . Let's assume the processing of the 10,001st record; sites A and B will use `processRecord` to classify their own record fragment of the 10,001st record. Suppose, A is unconfident (making the condition on line 5 of Algorithm 1 true), and B is confident for this record and marks the same condition as False. This record is the first record after *batch-1* and doesn't belong to *batch-1*. The condition specified on line 6 becomes false, and on line 8 turns out to be true in the same algorithm. Therefore, site A will use the values sent as feedback from T_{AB} to calculate the left side of Equation 3.4. By this, site A tries to predict the probability of B being confident. A, for example, calculates this value to be 0.7, which is greater than 0.5; in this case, A will assume that B is confident for this record and will restrict the transmission of this record (10,001) to T_{AB} , making the else part on line 12 of Algorithm 1 true. This would save the transmission cost because if A would have sent this record to T_{AB} and B, being confident, would not have its record fragment. As a result, T_{AB} would not process this record (according to `forwardTroubleRecord` of Algorithm 2) as it will receive a record fragment from only one site. But, in the other case, if the calculated value was 0.3, which is less than 0.5. The condition specified on line 10 of Algorithm 1 would turn out to be true. A will assume that B is also unconfident about this record and will send the record fragment to T_{AB} . Here, no savings would be earned for communication, and also, there will be no effect on accuracy.

The reason for setting the peer confidence threshold at **0.5** is that an unconfident site, when estimates that a peer site can make a confident prediction above random

chance, then it should restrict the transmission of the record to the trouble site. The aggregation site will determine the class value for such a record based on the site's perception of a confident prediction from its sister peer site. If the site's perception of its peer was incorrect, then that would refrain processing of the record at the trouble site, ultimately resulting in the absence of classification results from the trouble site, which may have a detrimental effect on accuracy. Ultimately, setting peer confidence threshold at a modest value of **0.5** ensures that communications overhead is minimized by preventing transmissions involving assessments with a probability of being confident greater than random chance rather than requiring a higher hurdle, such as a value in the range $[0.5, 1.0)$. One consequence of setting it at the random chance value is that there is a risk of not identifying cross-terms since such records are not dealt with at the trouble site. In effect, the setting of the threshold represents *a trade-off between classification accuracy and communications overhead*.

Each trouble site runs a procedure `forwardTroubleRecord` of Algorithm 2, which shows that if it receives record fragments from all its source sites (paired source sites), then it processes that record. In other words, if all of the trouble site's source sites agree that a record is a trouble, then the trouble site processes that record. While if it receives the record from only one site, it uses that particular record for updating existing statistics or computes new values in case they do not exist. As a result, the calculated feedback data or statistics are transmitted to the source sites for the first time at the end of *batch-1* and then incrementally updated on a record-by-record basis in order to keep statistics current.

All the sites in the network must send their local classification with the confidence value to the aggregator, which will compute the final prediction depending upon the aggregation method used; the details of aggregation methods have been provided in further sections. The role of the aggregator has been elucidated in the procedure `aggregateClassification` of Algorithm 3. The role of a source site has been

Algorithm 1: PODSM: Part I - Primary site role.

```

1 Procedure processRecord (site, record) :
2   classify record at site to get the classification-result;
3   aggregateClassification (site, record-key, classification-result,
4     t-site-forwarding) ;
5   foreach t-site that site may forward to do
6     if classification-result confidence < threshold from site to t-site then
7       if record belongs to batch-1 then
8         forwardTroubleRecord (t-site, record) ;
9       else if feedback data received from t-site for  $(X = x_1, x_2, \dots, x_n) \neq 0$  then
10        calculate the conditional probability of another paired site being
11        confident given the present site is unconfident for  $(X = x_1, x_2, \dots, x_n)$ ;
12        if the above estimated classification-result confidence of other
13        site < 0.5 then
14          forwardTroubleRecord (t-site, record) ;
15        else
16          quit processing the current record and resume with the next
17          record;
18        end
19      else
20        forwardTroubleRecord (t-site, record) ;
21      end
22    end
23  end

```

Algorithm 2: PODSM: Part II - Trouble site role.

```

1 Procedure forwardTroubleRecord (t-site, record-fragment) :
2   if t-site has received a record-fragment from all source sites then
3     processRecord (t-site, merged-record) ;
4   else if record belongs to batch-1 then
5     add record-fragment to a size-limited buffer at t-site;
6     calculate historic statistics when  $(X = x_1, x_2, \dots, x_n)$  where X is the
7     feature set;
8     transmit it to the source sites as feedback data;
9   else
10    update the existing historic statistics if they exist otherwise, calculate
11    new values and transmit it to the source sites as feedback data;
12  end

```

Algorithm 3: PODSM: Part III - Aggregator role.

```

1 Procedure aggregateClassification (site, record-key,
  classification-result, t-site-forwarding) :
2   log classification-result as received from site for record-key;
3   foreach t-site in t-site-forwarding do
4     if site forwarded record-key to t-site AND record-key has not been
       logged as unexpected from t-site then
5       | log that a result for record-key may be expected from t-site;
6     else
7       | log that a result for record-key is unexpected from t-site;
8     end
9   end
10  if all p-sites and expected t-sites have provided a classification-result
    then
11    | aggregate all classifications for record-key into a final-classification;
12  end

```

summarized in the form of a flowchart in Figure 3.4.

3.3.2 Illustration of transmission protocol

The aggregator has all the information required for classifying every record through the transmission mechanism described in Algorithms 1, 2, and 3. The completeness of this transmission protocol can be demonstrated by considering a hierarchy, where p_1 and p_2 are the two primary sites that classify their record segments r_1 and r_2 of record r with key k and produces c_1 and c_2 as their respective classification, and t is the trouble site with c_t representing its classification.

There can be *four* possible cases to elucidate the transmission process. The *first possible case* is when both the primary sites are confident. In this situation, both the sites classify their record segment with confidence values greater than the threshold value, so both of them p_1 and p_2 would send their classifications c_1 and c_2 to the aggregator without forwarding their record fragments to the trouble site. The sites notify the aggregator that they are not forwarding the record to the trouble site by setting

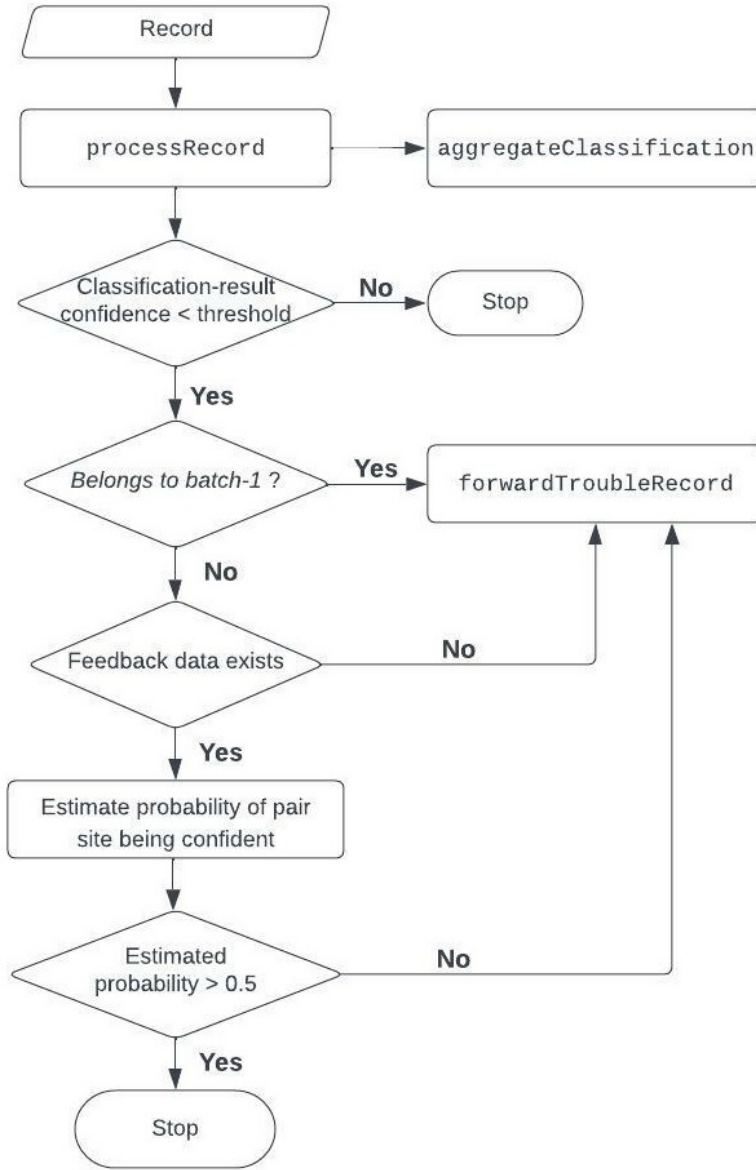


Figure 3.4: A flow diagram summarizing the role of the primary site in PODSM architecture.

t -site-forwarding = $\{t: 0\}$ on line 3 while calling `aggregateClassification` in Algorithm 1. When the aggregator starts processing each site's classification, as both the sites p_1 and p_2 sent $\{t: 0\}$ in the call; therefore, the record is logged as unexpected from t . Ultimately, the aggregator will process the classification received from sites (i.e., from both the primary sites and not expected from the trouble site) and produce the final classification. Algorithm 4 outlines the high-level procedure calls involved in this case.

Algorithm 4: Procedure calls for case when both p_1 and p_2 are confident.

```

1 processRecord( $p_1, r_1$ )
2   > aggregatorClassification( $p_1, k, c_1, \{t: 0\}$ )
3 processRecord( $p_2, r_2$ )
4   > aggregatorClassification( $p_2, k, c_2, \{t: 0\}$ )

```

In the *second case*, p_1 confidently classifies the record while p_2 is not confident in its classification, so in this case, if the record r belongs to *batch-1*, then t-site-forwarding will be $\{t: 1\}$ for p_2 and $\{t: 0\}$ for p_1 , here only p_2 will send its record fragment to trouble site. While if the record does not belong to *batch-1*, then p_2 will use statistics or feedback data to estimate the probability of confidence of p_1 . If the estimated probability of confidence of p_1 by p_2 is less than the threshold, the outcome will be again $\{t: 1\}$ for p_2 and $\{t: 0\}$ for p_1 , but if the estimated probability is greater than the threshold, the t-site-forwarding will be $\{t: 0\}$ for both p_1 and p_2 (i.e. p_2 will estimate that p_1 is confident and will not transmit its record, thus saving communication). If t-site-forwarding is $\{t: 0\}$ for both the sites, the situation will be the same for the aggregator as in the first case, while if p_2 sends its record fragment to t toggling the condition to $\{t: 1\}$. The trouble site t will then receive the record segment from only one site; therefore, t will not be processing the record as both the source sites differ in their decision. As a result, the record turns out to be useless from an operational view at the trouble site, but with the application of PODSM, this record would be added to the buffer and will be used for the calculation of statistics, thus logging the classification to be unexpected from t . The aggregator will produce the final classification from the classification of both sites and will not wait for t . Algorithm 5 outlines the procedure calls made when record r belongs to *batch-1* or if the estimated probability of p_1 being confident by p_2 is less than the threshold; for other cases, Algorithm 4 will be used.

The *third possible case* is when the primary site p_1 is unconfident, and p_2 is confident, which is similar to the second case except that the role of the primary site gets

Algorithm 5: Procedure calls for case when p_1 is confident and p_2 is not confident.

```

1 processRecord( $p_1, r_1$ )
2   > aggregatorClassification( $p_1, k, c_1, \{t: 0\}$ )
3 processRecord( $p_2, r_2$ )
4   > aggregatorClassification( $p_2, k, c_2, \{t: 1\}$ )
5   forwardTroubleRecord( $t, r_2$ )

```

interchanged; therefore, this case does not need further discussion.

The *final and fourth case* is when both the primary sites are unconfident. If the record belongs to *batch-1*, then both the sites will transfer their respective record fragment to the trouble site. Thus, t-site-forwarding will be $\{t: 1\}$ for both, ultimately logging the expectation of classification results from the t-site. After receiving the record from both sites, the trouble site will process the merged record and label the t-site as active for the classification result for this particular record. Finally, the aggregator will generate the final classification based on the classification received from both the primary sites p_1 and p_2 , plus it will also wait for the classification from t. Algorithm 6 demonstrates the procedure calls used in this particular case. On the contrary, if the record does not belong to *batch-1*, both the primary sites will estimate the probability of confidence of the other paired site using the historic statistics. If the estimated probability value is greater than the threshold, the transmission of the record will be skipped by that unconfident primary site. In the scenario where both sites wrongly estimate their paired site to be confident, both the unconfident primary sites will not send the record to the trouble site following Algorithm 4. If one of the sites made a wrong estimation and the other made the correct prediction, the second and the third cases will be used depending on the confidence status of the primary sites. The last scenario would be if both the sites correctly predict the low confidence of each other; in this case, Algorithm 6 will be used. For all cases, incorrect predictions may affect accuracy.

Algorithm 6: Procedure calls for case when both p_1 and p_2 are not confident.

```

1 processRecord( $p_1, r_1$ )
2   > aggregatorClassification( $p_1, k, c_1, \{t: 1\}$ )
3   forwardTroubleRecord( $t, r_1$ )
4 processRecord( $p_2, r_2$ )
5   > aggregatorClassification( $p_2, k, c_2, \{t: 1\}$ )
6   forwardTroubleRecord( $t, r_2$ )
7   /* t has received fragments from all source sites, and therefore processes r
   */
8   > processRecord( $t, (r_1 + r_2)$ )
9   > aggregatorClassification( $t, k, c_t, \{\}$ )

```

3.3.3 Aggregation methods

The final classification outcome by the aggregator can be produced based on several methods. The first one is *maximum confidence* which has been adopted as the standard aggregation method in the experiment assessment section below. Under maximum confidence aggregation, a site with the highest classification confidence is the victor (Park et al., 2001), and the aggregator chooses the winner site's class as the final selection. The other aggregation method is *voting*. Under this approach, the aggregator's outclass is based on majority votes from all the sites, and the final confidence of classification is the mean confidence of all of them (Skillicorn & McConnell, 2008). The third aggregation method is *stacking* (Parker et al., 2012), where the classification results from all the sites are used as input to another stream classifier; however, these stream classifiers may or may not use the same algorithm as the base classifiers. These stream classifiers generate a new classification with a confidence value. The impact of using different aggregation methods is depicted in the next section.

3.4 Experimental Evaluation

The source code for state-of-the-art work and PODSM has been made available in a GitHub repository; refer to Appendix C for details.

3.4.1 Experimental setup

All experiments were performed with OpenJDK 1.8.0 on a 64-bit Windows running on a 4×3.40 GHz Intel Core i5 CPU with 16 GB of memory. We have used *Adaptive Random forest* (ARF) (H. M. Gomes et al., 2017) classifiers at trouble sites while *Naïve Bayes classifiers* (Webb, 2010) are used at the primary sites, and the default method for aggregating the classification results has been set to maximum confidence. Several configuration parameters were fixed throughout the implementation phase in HDSM (Denham, 2019). They were *size-limit for all windows*, *all threshold values for creation and removal monitors*, *Hoeffding bound (δ)*, and *smoothing factor for agreement threshold (θ)*. Additionally, in PODSM, an invariable known as *batch-1* was given a constant value depending upon the size of the dataset such that there are enough records for the calculation of statistics and also to ensure that these statistics are applied to a significant number of records. For example, the value of *batch-1* for the HIG dataset was 5000, while for FLT, it was 10000. Typically, the value of *batch-1* was set to half the size of the complete dataset for most of the datasets. The total record count for each dataset has been listed in Table 3.2. Trouble Factor (TF) is another user-configured parameter in the HDSM architecture that determines the proportion of the data stream that will be transmitted to a trouble site in the form of trouble records. In other words, it allows users to control the volume of data processed at a trouble site. As such, a trouble factor equal to the number of features at a primary site would represent a trouble site capable of receiving as much data as a primary site (refer Denham (2019) for more details). The experiments were configured for TF equal to 1, 1.5, and 2

times the maximum number of features taken across the set of primary sites. A blend of real-world and synthetic datasets were used to evaluate the PODSM algorithm's performance. The reason for selecting real-world datasets is that they represent data streams that may be vertically distributed in practice as they indicate the presence of cross-terms between the distributed feature sets. The experiments were performed on nine real-world datasets, namely, Occupancy (OCC), NASA FLTz (FLT), HIGGS (HIG), Gas Sensor Array Drift (GAS), Gesture Master (GES), Skin NonSkin (SKIN), Parkinson (PARK), Fog release (FOG), Sensorless Drive Diagnosis (SDD). Moreover, two synthetic stream datasets were also selected for assessing the performance under known concept drifts; *incremental and gradual drift* (discussed in Section 1.2). The synthetic stream datasets, *RBF-M* and *SEA-G*, were based on the Radial Basis Function generator (RBF) (Bifet & Kirkby, 2009), and SEA generator (Street & Kim, 2001).

Table 3.2 describes the datasets in terms of feature count, number of primary sites, classes, and records count and lists the number of features available per site. The column "Stream" provides streaming information about the dataset. Some datasets were an actual time series of events in their correct time-based order, so could be evaluated in an online setting in that original order. But, some were not actually time series, so were organized prior to evaluation to avoid issues like where the original ordering grouped instances by class. As such, in some cases, the dataset consists of long periods of a single class where shuffling was needed for meaningful assessment, while for the SDD dataset, there were 11 classes producing several streams, so these streams were interleaved for preserving time-series progression within them. Interleaving was done by drawing records in order from each stream until all records from all the streams were exhausted, and the selection of the next stream was made randomly. Finally, for synthetic datasets, the "Stream" column notes the kind of simulated shift. FLT data were retrieved from NASA's data portal (N. Oza, 2011), and the rest of the real-world data were retrieved from UCI Machine Learning Repository (Dua & Graff, 2017) while

Table 3.2: Properties of datasets used for performance evaluation.

Dataset	Feature count	Feature per site	P-site count	Class count	Record count	Stream
FLT	20	1-3	9	2	25034	Shuffled
OCC	3	1	3	2	20560	Shuffled
HIG	19	3-4	5	2	10000	Stream
GAS	128	8	16	6	13910	Stream
GES	8	1	8	6	63196	Shuffled
SKIN	3	1	3	2	50000	Shuffled
PARK	6	1	6	2	26731	Shuffled
FOG	9	1	9	3	38774	Shuffled
SDD	48	4	12	11	58509	Interleaved
SEA-G	3	1	3	2	100,000	Gradual drift
RBF-M	10	1	10	5	50,000	Moderate Incremental Drift

the synthetic stream datasets SEA-G and RBF-M were generated with MOA (Bifet et al., 2010).

3.4.2 Performance assessment metrics

The *accuracy* and the *Total Transmission (TT)*, both reported in terms of percentage (%), were used as metrics to evaluate the performance of the PODSM algorithm. The accuracy was evaluated using the confusion matrix produced as part of the classification process; also, it was calculated prequentially. In *Prequential or Interleaved Test-then-Train* evaluation as stated in Section 2.3, each instance is first utilized for testing the model before further being used for training; this way, testing is always done on unseen examples. TT reports the data transmission in the form of records between all the sites. Transmission of site classifications to the aggregator is not considered, as the transmission of class labels and confidence values from the sites would be the same for all the experiments, being a much less significant cost in comparison to record transmission. The resource time is reported in terms of Mean Time on Critical Path (MTCP) and Mean Time Between Completions (MTBC). *MTCP* measures the classification interval for a simple record on average, and it is the cumulative CPU time

Table 3.3: Explanation of metrics used for performance assessment.

Metric	Explanation
Accuracy	Classification accuracy.
TT	Total data transmission to trouble sites.
MTCP	Mean Resource Time (CPU) to generate final classification on critical Path.
MTBC	Mean Resource Time (CPU) between successive classification completions.

taken to process a record over the longest-running sequences of classifiers averaged over all the records. While *MTBC* measures the mean difference between the execution times of consecutive records, provided that each classifier can start processing a record only after it has finished processing the previous record and also after all its source sites have processed the record. MTCP and MTBC are expressed in nanoseconds. Table 3.3 provides a summarised definition for all the assessment units.

3.4.3 Performance comparison

The following section presents the comparative study between state-of-the-art methods and PODSM. The efficacy of PODSM would be assessed by comparing its experimental performance to the most closely related work, *distributed ensembles*, *Parks' algorithm*, and *HDSM* (discussed in Section 2.6). The results obtained by applying the PODSM algorithm demonstrate its ability to achieve a substantial drop in communication while maintaining nearly the same accuracy levels. The values for these metrics were computed without taking *batch-1* into account as *batch-1* was only used for calculating the initial values of statistics, and in actuality, there was no effect when applying this algorithm on *batch-1*. Tables 3.4 and 3.5 provide a comparative study for accuracy and TT between all related methods and PODSM. As the distributed ensembles involve nil communication between the distributed sites, their values have not been included in Table 3.5. For almost all the datasets, HDSM proves to be a better mechanism

Table 3.4: Accuracy comparison between state-of-the-art methods and PODSM.

Dataset	D.Ensem.	Parks'	HDSM			PODSM		
			TF 1x	TF 1.5x	TF 2x	TF 1x	TF 1.5x	TF 2x
FLT	68.46	68.46	81.18	87.37	87.62	80.54	87.01	87.62
OCC	79.01	82.00	88.67	88.78	92.69	82.34	88.33	91.19
HIG	52.84	53.14	53.90	54.00	54.48	54.24	53.58	54.72
GAS	38.40	38.40	63.04	73.05	81.49	60.95	70.80	81.40
GES	62.91	62.91	63.08	63.01	63.29	62.97	63.07	63.29
SKIN	94.59	96.81	97.78	96.68	94.83	95.97	95.90	94.83
PARK	72.03	72.01	79.53	92.52	99.59	77.46	92.52	99.59
FOG	68.20	68.20	85.38	88.27	89.20	84.40	87.90	89.28
SDD	46.00	46.00	86.34	95.83	98.00	71.60	87.86	97.68
SEA-G	76.14	79.40	75.95	75.76	75.93	75.60	75.76	75.93
RBF-M	31.64	31.64	43.11	45.72	52.76	41.86	47.23	52.69

for learning the cross-terms, thus enhancing accuracy and involving more volume of communication. Results reported in Tables 3.4 and 3.5 depict PODSM's efficiency in declining communication while obtaining nearly the same accuracy except for two adverse cases: *SDD* and *GES*. In the case of *SDD*, the drop in accuracy is much higher when compared to savings in communication, whereas when TF equals 1.5 for *GES*, a contradictory effect on communication can be seen after applying PODSM. The highlighted figures represent the best case of improvement by PODSM over HDSM.

For the OCC dataset, when TF equals 1, the accuracy drops by 7%, but on the other hand, the algorithm achieves a colossal reduction in communication of approximately 42%. Table 3.6 summarizes the performance for the OCC dataset in terms of other metrics; *precision*, which measures the exactness, and *recall*, which is the measurement for completeness.

However, it can be witnessed that in some cases (SEA-G when TF is 1.5x, 2x), even after applying algorithm PODSM, communication and accuracy remains unchanged. Figure 3.5 depicts the efficiency of PODSM by showing its comparison with HDSM for the best case of TF for each of the datasets. This clearly reflects a significant cost

Table 3.5: Comparison between state-of-the-art methods and PODSM in terms of communication.

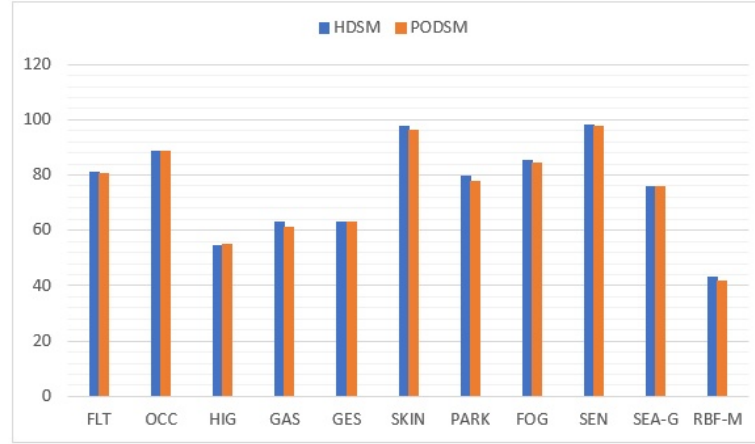
Dataset	Parks'	HDSM			PODSM		
		TF 1x	TF 1.5x	TF 2x	TF 1x	TF 1.5x	TF 2x
FLT	266.34	425.31	632.46	669.66	369.47	616.74	669.66
OCC	199.45	99.31	149.68	307.20	57.70	108.12	220.78
HIG	218.90	218.12	342.14	525.52	181.24	307.36	343.38
GAS	205.19	543.31	850.77	952.87	419.05	707.98	892.85
GES	200.11	297.97	276.86	267.28	279.56	353.11	267.28
SKIN	199.90	100.65	76.35	13.76	36.95	57.37	13.76
PARK	223.30	204.99	311.04	398.46	150.34	311.04	398.46
FOG	207.92	341.09	586.54	769.69	230.65	522.34	766.43
SDD	207.38	334.88	421.13	493.83	294.28	378.64	463.17
SEA-G	199.54	51.34	16.38	6.74	32.21	16.38	6.74
RBF-M	196.70	476.33	688.02	1026.22	327.35	684.55	1023.76

Table 3.6: Performance evaluation for OCC dataset in terms of Precision and Recall.

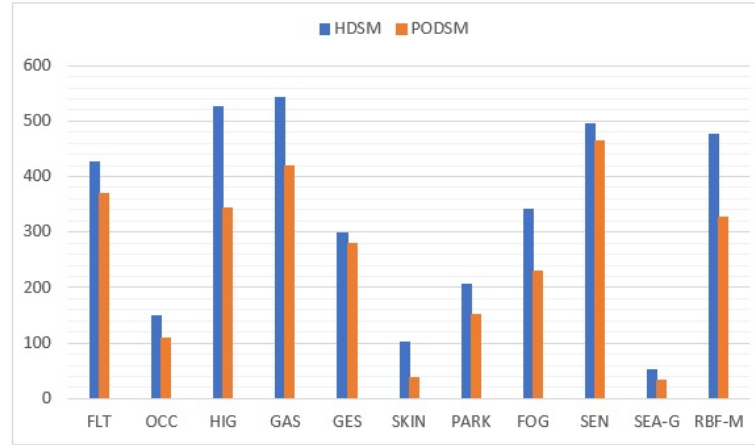
TF	HDSM				PODSM			
	Precision		Recall		Precision		Recall	
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1
1x	0.93	0.74	0.92	0.77	0.85	0.67	0.93	0.46
1.5x	0.93	0.75	0.93	0.76	0.93	0.74	0.92	0.75
2x	0.95	0.85	0.96	0.82	0.93	0.85	0.96	0.75

reduction in communication with a marginal effect on accuracy. Figure 3.5a shows the comparison of accuracy between HDSM and PODSM, whereas, Figure 3.5b presents the comparison in terms of communication involved.

Table 3.7 draws the comparison between HDSM and PODSM in terms of resource time when TF equals 1. In almost all the cases, an improvement in resource time is observed except for *GES* and *FLT* (for MTBC). Considerable reduction in resource time has been obtained while maintaining the same accuracy levels, with SDD and RBF-M showing remarkable efficiency.



(a) Accuracy



(b) Communication

Figure 3.5: Significance of PODSM shown through graphical representation in terms of accuracy and communication, a) demonstrates how PODSM impacts accuracy while b) presents the communication comparison between HDSM and PODSM for the best case of TF.

As discussed in Section 3.3.3, there can be different ways of combining the classification results from all the sites at the aggregator site. The effect of the choice of method to produce the final classification on accuracy and TT for the SKIN dataset has been depicted in Figure 3.6, which has three subgraphs, one for each case of TF. SKIN is one of the datasets that showed good communication cost improvement while not affecting accuracy much. In the subgraph for TF equals 1, a major decline in transmission has been achieved for all methods with marginal effects on accuracy. When TF is 1.5x, and the aggregation method is maximum confidence, transmission witnesses a drop

Table 3.7: Resource time comparison between HDSM and PODSM.

Dataset	HDSM		PODSM	
	MTCP	MTBC	MTCP	MTBC
FLT	1.80E5	7.05E4	1.67E5	7.06E4
OCC	7.63E4	6.63E4	6.34E4	3.84E4
HIG	1.45E5	6.43E4	1.40E5	4.69E4
GAS	1.85E5	2.63E4	1.67E5	2.17E4
GES	1.33E5	1.31E4	1.38E5	1.50E4
SKIN	3.12E4	2.49E4	2.01E4	8.11E3
PARK	9.47E4	5.34E4	7.86E4	2.23E4
FOG	1.47E5	7.92E4	1.23E5	6.21E4
SEN	1.43E5	6.61E4	1.20E5	1.71E4
SEA-G	4.58E4	2.96E4	4.15E4	2.54E4
RBF-M	2.08E5	6.35E4	1.65E5	1.74E4

Table 3.8: Resource time comparison for different aggregation methods.

TF	HDSM						PODSM					
	MaxConf		Voting		Stacking		MaxConf		Voting		Stacking	
	MCTP	MTBC	MCTP	MTBC	MCTP	MTBC	MCTP	MTBC	MCTP	MTBC	MCTP	MTBC
1x	3.12E4	2.49E4	2.88E4	2.19E4	1.07E5	9.25E4	2.01E4	8.11E3	1.74E4	7.35E3	1.00E5	8.87E4
1.5x	2.54E4	1.86E4	5.25E4	4.52E4	1.14E5	9.56E4	2.21E4	1.54E4	4.91E4	4.28E4	1.12E5	9.46E4
2x	1.36E4	5.69E3	5.46E4	4.78E4	1.36E5	1.06E5	1.28E4	5.50E3	5.14E4	4.48E4	1.25E5	9.63E4

of 24.86% with a 0.81% decrease in accuracy; on the other hand, for the other two methods, there seems to be less or no effect. Similarly, for TF 2x, voting shows a slight decline in transmission with a minor effect of 0.17% on the accuracy, while the other two method exhibits no change. The impact of using different aggregation methods on resource time has been summarized in Table 3.8, which shows substantial improvement in performance in terms of resource time; bold values represent the best case. This relative analysis portrays PODSM's efficiency in achieving a substantial decline in communication and resource time while maintaining nearly the same level of accuracy for all the methods.

A relative view has been drawn among HDSM and PODSM in Figures 3.7 and 3.8. It displays *accuracy*, *data transmission volume*, and *resource time* timeline. *Data*

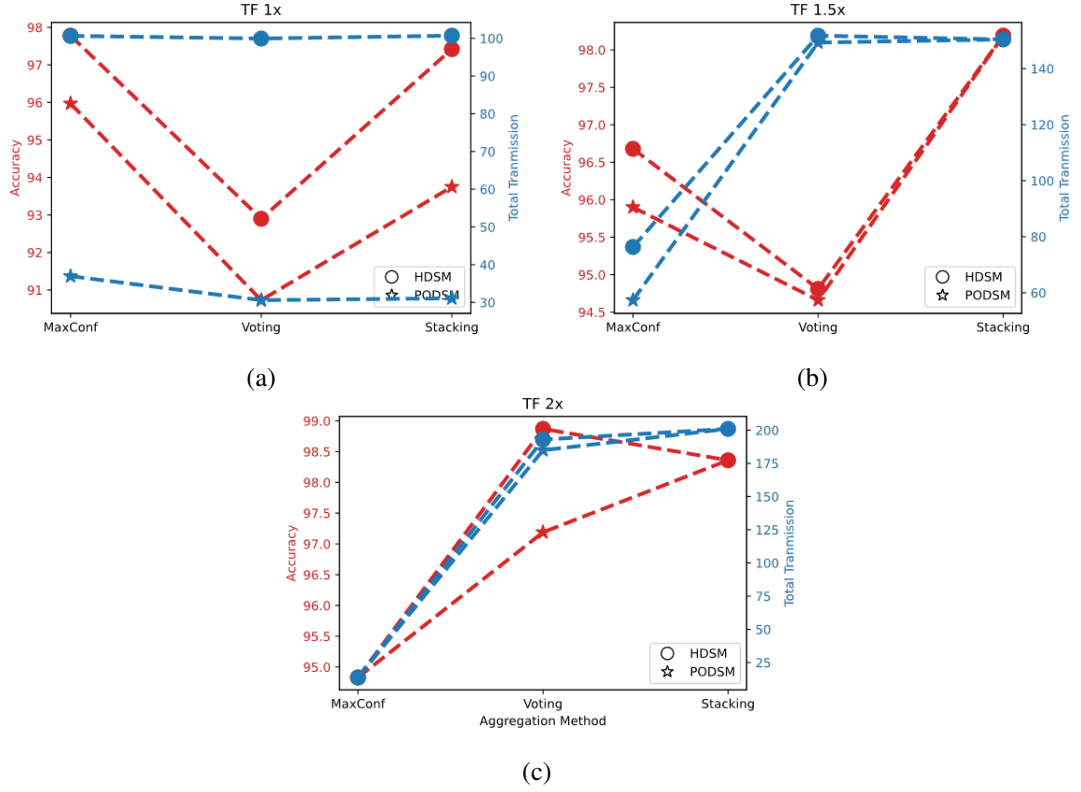


Figure 3.6: Accuracy and transmission comparison between HDSM and PODSM for all aggregation methods when A) TF is 1x, B) TF is 1.5x, and C) TF is 2x, for the SKIN dataset.

transmission volume is the record feature communication and is expressed as the proportion of the total volume of data in the dataset. *Total transmission* under the Data Transmission curve represents the volume of transmission between all sites, while *maximum transmission* to one trouble site expresses the maximum volume of transmission to any trouble site. Figure 3.7 presents the timeline for HDSM; it can be seen that after processing approximately 9000 records, a trouble site [2, 3] is generated. The addition of trouble site [2, 3] into the system captures the cross terms between sites 2 and 3, thereby rising accuracy levels while at the same time transmission levels also show a spike. Thus, higher accuracy is achieved at the expense of data transmission. Another trouble site [1, [2, 3]] gets created due to high agreement between the sites, thus capturing the cross terms. After a while, the recently added trouble site is removed and blacklisted because the cross terms for which it was created have disappeared.

Throughout the experiment, the trouble site [2, 3] remains active, revealing the presence of cross terms and playing a consistent role in improving the accuracy. The total transmission and resource time shows a spike whenever a trouble site is created, while maximum transmission to a trouble site maintains a steady trend.

Meanwhile, visualizations obtained after applying algorithm PODSM are represented in Figure 3.8, which reveals that the same trouble site [2, 3] is created at the same scales, showing an increase in accuracy and communication. However, in this case, maximum transmission to the trouble site shows a declining trend after some time because, at this stage, the source sites have started using the statistics sent by the trouble sites to predict the probability of confidence of the peer site, hence reducing communication and resource time. The trouble site [2, 3] gets removed due to a decline in the agreement between the sites; from this point, trouble site transmission turns out to be null until a new trouble site is created. The same recurring behavior can be witnessed as maximum transmission after reaching a height maintains a relatively balanced level for some time because the transmission during this period is used for calculating the statistics. When these statistics or feedback are used, it induces a decreasing trend in transmission. The trouble site [1, 2] gets removed since its utilization is dropped due to the absence of cross-terms, so this trouble site gets removed. During the ending phase, cross-term presence triggers the creation of a trouble site [1, 3]. Therefore, it can be noticed that after applying PODSM, the sequence of creation and removal of trouble sites differs from HDSM.

A final distinction has been drawn between centralized experiment and PODSM with maximum confidence as the aggregation method for all cases of TF. Relative ranks calculated for accuracy and communication averaged across eleven datasets have been reported in Figure 3.9, where points lying near to origin represent better ranks. However, a centralized experiment wins the race for accuracy but lags in maintaining the trade-off between accuracy and communication (calculated as the distance from the origin). In

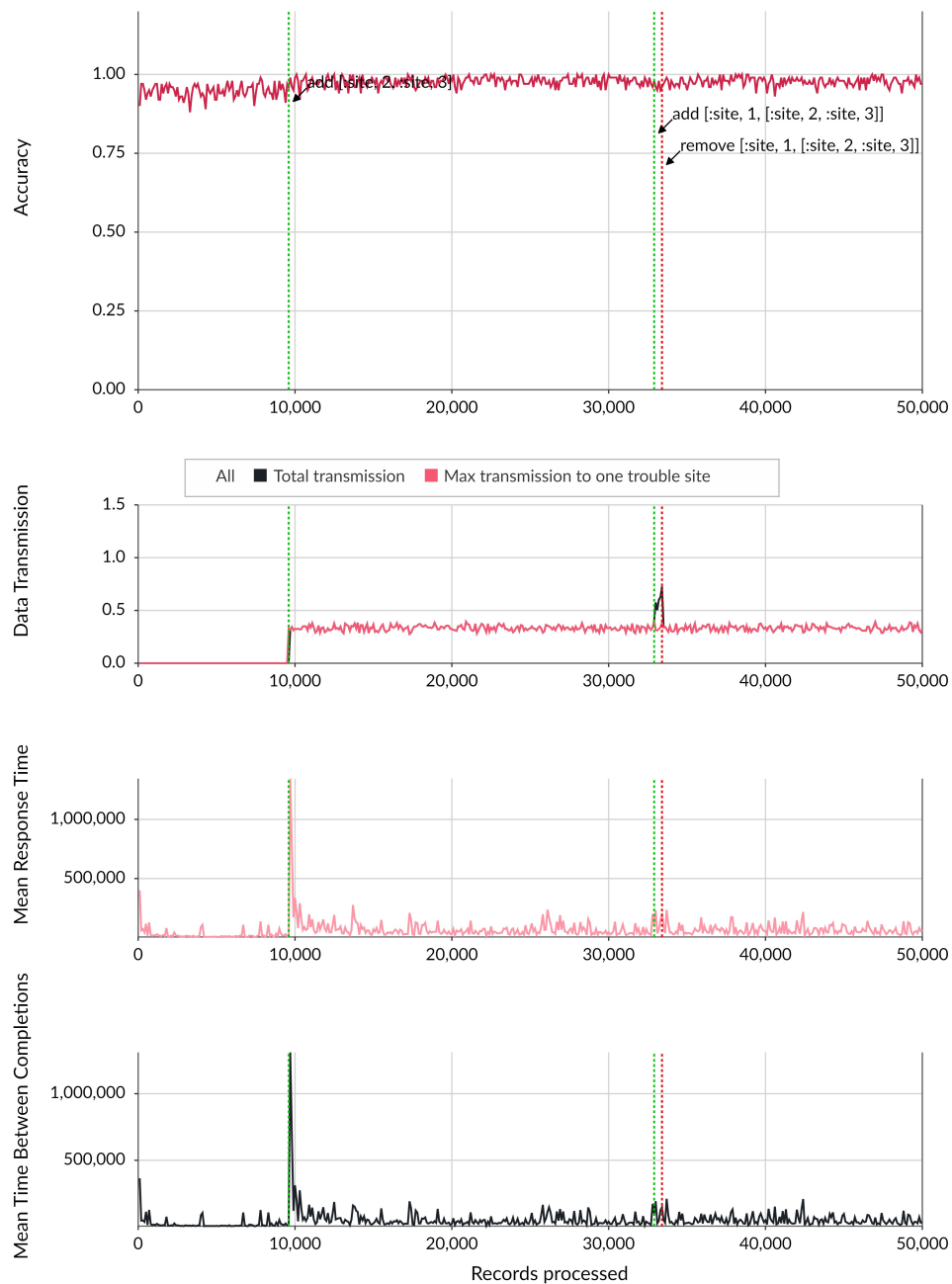


Figure 3.7: Experimental timeline for HDSM consisting of four graphs, the first graph shows the accuracy trend, transmission usage is presented in the second graph while the last two graphs show the resource time usage.

contrast, all cases of PODSM show better performance in building the balance between the two.

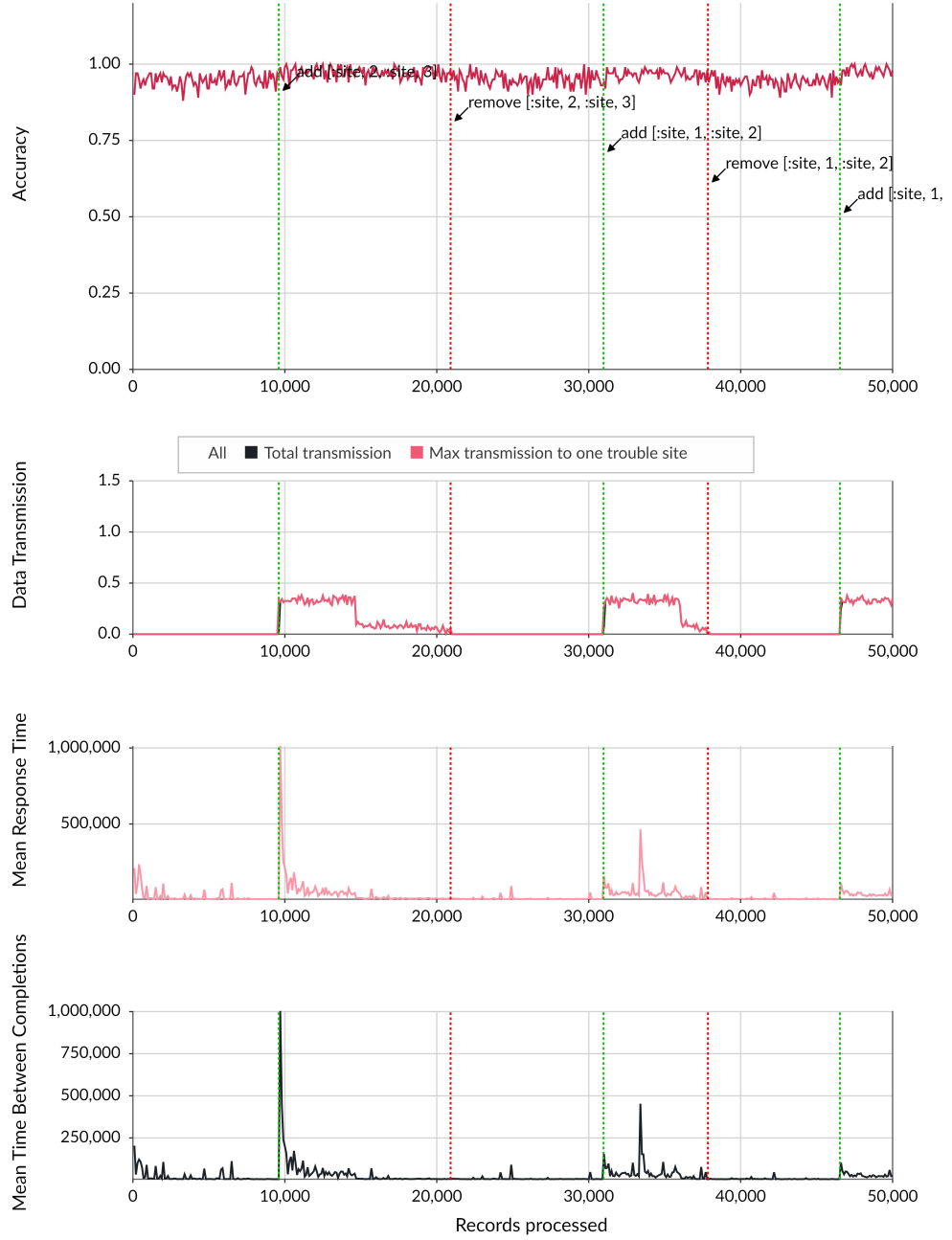


Figure 3.8: Timeline visualizations for PODSM experiment showing accuracy trend, and transmission and resource time utilization.

3.4.4 Parameter sensitivity analysis of PODSM

Along with configuring the experiments to different values of TFs, the sensitivity of PODSM was evaluated on two other parameters; *size of batch-1* and *periodicity of*

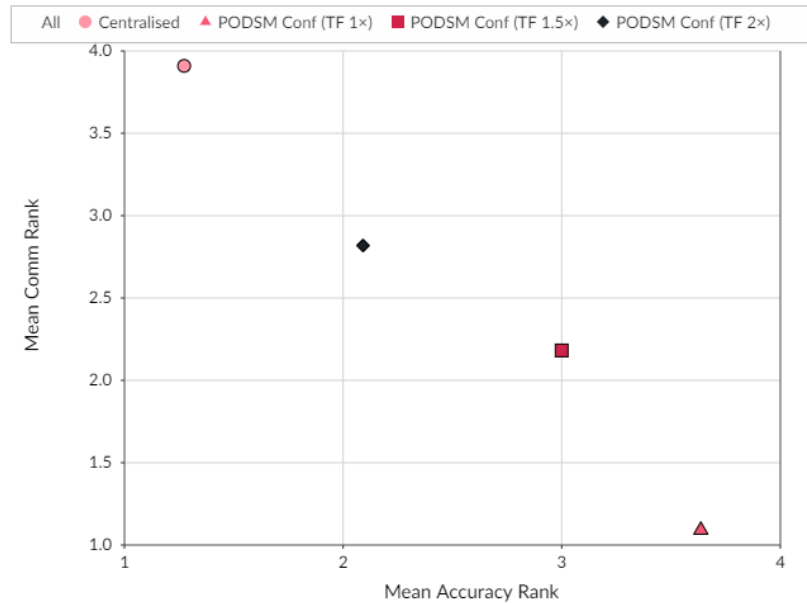


Figure 3.9: A ranking comparison depicting competency to maintain the trade-off between accuracy and communication between centralized experiment and PODSM configured for TF 1x, 1.5x, and 2x.

updating the statistics. Differing values were tested for both the parameters; when the value of one parameter was changed, the other parameters were stabilized to their standard value (as used in all the previous experiments). The size of *batch-1* was increased for some datasets while it was decreased for others. For instance, in the case of the HIG dataset, the standard size of *batch-1* used for all the previous experiments was 5000, which was decreased to 2500 for examining the influence of the size of *batch-1* on accuracy and communication. The results in Table 3.9 shows the impact of changing the size of *batch-1* (**Increased / Decreased**) on accuracy and communication. A direct relationship between batch size and communication can be witnessed from the results.

Once the statistics get calculated at the end of *batch-1*, they are updated on a record basis (as stated above under the default configuration of PODSM). To check this parameter's sensitivity, the frequency of refreshing these statistics was switched from a record basis to situations when any change in the hierarchical structure of trouble sites is

Table 3.9: Impact of change in the size of *batch-1* on accuracy and communication.

Dataset	<i>batch-1</i> size	Accuracy			Communication		
		TF 1x	TF 1.5x	TF 2x	TF 1x	TF 1.5x	TF 2x
FLT	Dec	79.81	86.36	86.95	356.62	598.03	676.31
OCC	Dec	80.61	84.38	90.31	39.65	85.28	223.52
HIG	Dec	53.93	54.90	54.04	113.39	277.37	345.49
GAS	Dec	63.23	72.14	80.28	357.34	552.14	739.71
GES	Inc	63.18	63.09	63.24	295.38	253.09	186.49
SKIN	Dec	95.21	95.26	94.86	18.74	32.78	14.24
PARK	Inc	77.40	93.37	99.61	163.91	311.66	398.56
FOG	Dec	82.61	87.14	89.09	200.07	460.79	747.77
SEN	Inc	72.06	88.36	98.07	375.33	626.71	478.94
SEA-G	Dec	75.90	78.46	78.54	43.27	54.54	60.61
RBF-M	Inc	42.88	47.45	53.11	382.47	691.28	1036.11

seen, that is, on addition or removal of a trouble site. The readings in Tables 3.4 and 3.5 were obtained when the probabilities are updated on a record basis after *batch-1*, while Table 3.10 presents the results when the probabilities are only updated when there is some change in the structure, that is only when a trouble site is added or removed. It can be seen from the results presented in this table that a further decrease in communication is achieved, compromising accuracy levels marginally.

Results in Tables 3.9 and 3.10 highlights *the trade-off between classification accuracy and communication overhead*; therefore, the size of *batch-1* and the periodicity of updating probabilities can be adjusted to maintain a balance between the two.

3.5 Applicability and Limitations of PODSM

An essential characteristic of PODSM lies in its flexibility, which can provide fast and accurate classification results using less communication and computation. The backbone of the algorithm is to learn heuristics using historic data and employ them for new incoming records, thus saving resources in reference to communication and

Table 3.10: Effect of changing the frequency of updating the statistics.

Dataset	Trouble Factor					
	1x		1.5x		2x	
	Accuracy	Comm	Accuracy	Comm	Accuracy	Comm
FLT	80.80	390.83	87.20	615.24	87.62	669.66
OCC	84.46	38.41	88.57	120.08	92.31	259.13
HIG	54.32	219.70	53.48	399.16	55.24	445.56
GAS	60.49	474.16	65.58	719.24	80.90	980.16
GES	62.97	279.58	63.07	353.13	63.29	267.28
SKIN	97.62	39.94	96.55	69.14	94.83	13.76
PARK	75.39	170.24	92.52	311.04	99.59	398.46
FOG	84.59	240.67	88.17	561.52	89.33	768.16
SEN	77.40	329.29	92.01	423.65	97.91	497.47
SEA-G	78.24	55.52	75.76	16.38	75.93	6.74
RBF-M	40.78	345.99	45.80	691.91	52.25	1023.48

time. This probability calculation method has its primary application in resource-restricted heterogeneous frameworks. The development of IoT applications obliges the conjoint servicing of heterogeneous data streams like traffic congestion models and video surveillance systems involving numerous video cameras.

The experiment evaluation revealed the potential of PODSM to enhance the performance of mining distributed data streams while maintaining a balance between *accuracy, communication, and computation resource time*. As mentioned before, Naive Bayes classifiers, which are lightweight algorithms, were used at the primary sites, and the focus of this algorithm has been on *decreasing communication and resource time*. Therefore, PODSM has the ability to strengthen the WSN consisting of distributed sensor nodes with limited capabilities, such as the one used to predict seismic activities like an earthquake or tsunami early warning system. However, it is a flexible algorithm that can be used with many datasets and other classification methods or may even be applied to other machine learning scenarios such as numeric regression.

It is worth discussing this algorithm's current constraints and the scenarios where

PODSM may not be suitable. Table 3.5 shows that in some of the cases, even after implementing the algorithm, the communication factor remains unchanged; it may be mainly due to the inefficiency of the dataset in learning sufficient statistics that can be applied to estimate the probability of confidence of paired site and the other case can be when there is substantial fall in accuracy. One way to combat these cases can be switching to other TF values.

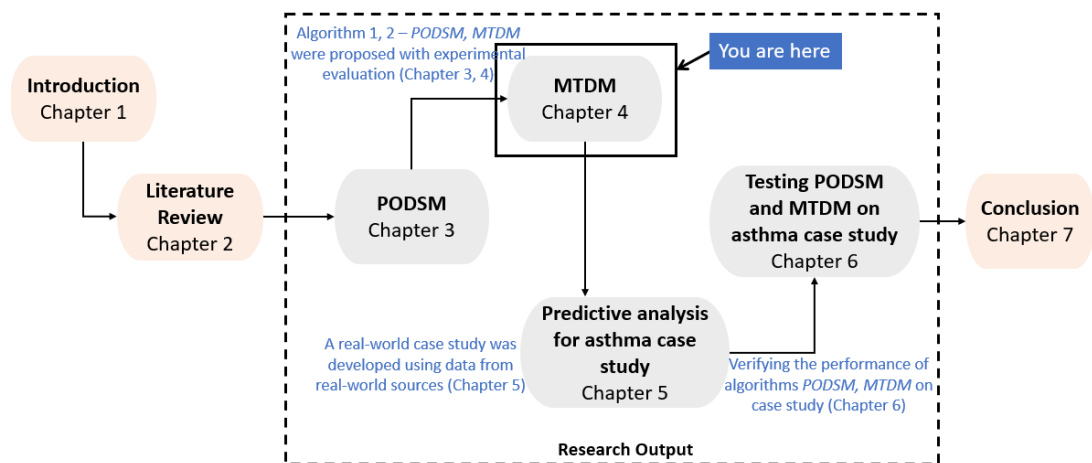
3.6 Chapter Summary

Mining data from a heterogeneously distributed architecture incurs huge communication costs. In this chapter, we presented a method to optimize communication overhead and resource time in vertically distributed data streams while maintaining similar accuracy level. This method used past behavior of the sites in the form of statistics to predict the confidence status of their peer site. For making these predictions, this method exploited Bayes' theorem to generate the probabilistic view of the sites. Our experimentation showed a significant drop in communication and an improvement in resource time with minimal effect on accuracy was achieved for all the datasets except GES and SDD.

Chapter 4

Minimized Tree for Distributed Mining (MTDM)

This chapter introduces an algorithm known as Minimized Tree for Distributed Mining (MTDM) for defining a robust and scalable methodology to learn the relationship between the distributed sites efficiently. The algorithm having its basis in trees, has been described in depth, followed by the empirical assessment of the proposed algorithm on eleven datasets and a comparison with the closely related work.



Roadmap of the thesis.

4.1 Introduction

Increasingly, a number of organizations that aim to operate globally are turning distributed data processing into a necessity. A *distributed site/node* in a distributed environment does not know the views of other distributed sites. Thus, mining data from such an environment is a struggling job that becomes an uphill battle when these sites are heterogeneous; that is, the views of these sites are diverse. *In essence*, discovering knowledge from heterogeneous sources flooding data has become a research epicenter. The main hurdle experienced in such a scenario is the *lack of an efficient methodology for learning relationships between the sites*, specifically in a streaming environment. In fact, grasping this relationship with minimum communication from all the sites is a major challenge in DDM. This chapter presents an algorithm, ***Minimized Tree for Distributed Mining*** (MTDM), which facilitates the process of determining the relationships between various distributed sites/nodes with the help of a minimized tree. A *minimized tree* that forms the nucleus of this algorithm is used in learning the coupling between the vertically distributed nodes. After applying MTDM to vertically distributed data streams, a massive curtailment in resource time can be achieved with a beneficial influence on accuracy.

4.2 Problem Definition

In DDM, it is assumed that the dataset is dispersed over various physical sites or nodes. The ultimate motive is to obtain a global model with minimum possible transmission from each distributed site and use the least computational resources in the form of resource time while fulfilling the accuracy requirements. As stated previously, a trouble site is created after merging multiple source sites; in some cases (Park et al., 2001), it is formed by merging all the distributed sites. The fundamental goal behind creating a

trouble site is to assist in the classification process of trouble records, to recall *these are records for which a site is unconfident*.

Different methodologies can be used for merging the source sites, and even the number of sites combined to form a trouble site may also vary. In some cases, sites can be combined depending upon their physical location, i.e., neighborhood sites may be combined to form a trouble site. However, the optimal way to merge them would be based on their potential to capture the *cross-terms* present across the feature set seen at different sites.

Let us use an example to exemplify all these terms, suppose there is a video surveillance monitoring station that captures video streams from a different set of cameras. This would be an image classification problem aimed at classifying suspicious activities from normal activities to send warning messages against any misconduct. In this case, each camera has a different orientation, thereby capturing different views of the image. As such, each camera can be considered a distributed site, capturing its own view, which is different from other sites. In order to classify a video stream into suspicious or non-suspicious class, each distributed site would produce its classification depending upon its local model for images and would forward its classification result to the central site for the final classification outcome, thus eliminating the requirement to transmit the actual image data. There is a possibility that these primary sites may not be confident enough about their classification. Those circumstances would require classification results from the merged feature set due to the presence of cross-terms, thereupon revealing the relevance of a trouble site. The unconfident distributed sites would send their image fragment to the trouble site, producing the classification result using the combined image data from its source sites.

As stated earlier, the number of sites joined to generate a trouble site can vary. It is worth mentioning here again Parks' method proposed all the primary sites' linkage to a single trouble site, while in HDSM, sites were combined in a pair of two. For instance,

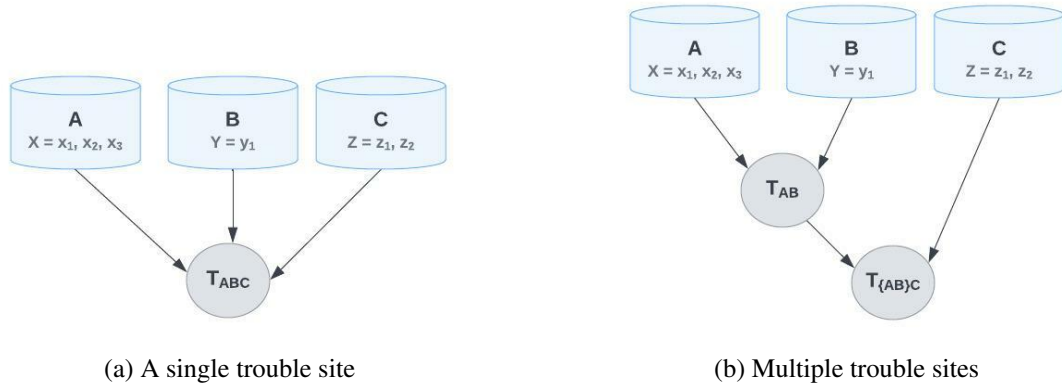


Figure 4.1: a) A single trouble site capturing cross-terms between all the distributed sites, b) Merging of distributed sites in pairs to capture cross-terms between subsets of distributed sites.

Figure 4.1 represents a vertically distributed scenario where; A , B , and C represent three distributed sites with feature sets X , Y , and Z , respectively. According to Parks' approach, A , B , and C would be merged into a single trouble site as represented in Figure 4.1a. Assume that there exists some indirect relationship between features x_i , y_i . The records for which sites A and B would be unconfident would be sent to the trouble site, while it's quite possible that Z may be confident for these records. Remember, a trouble site only processes a record when it receives the record fragment from all of its source sites; therefore, these record fragments sent by sites A and B won't be processed at the trouble site in the structure shown in Figure 4.1a. This would result in useless resource utilization and limit the involvement of the trouble site in the form of classification results. Hence, this structure will not be able to capture the cross-terms that exist only between sites A and B , whereas HDSM suggests the creation of a trouble site among a subset of primary sites (specifically in pair of 2) as shown in Figure 4.1b. Consequently, if only two sites are merged to form a trouble site, the number of records processed at the trouble site would increase. In such a way, the classification results would be called from the trouble site for a larger number of records, thereby providing more candidates for producing the final classification at the aggregation time.

Thus, the structure presented in Figure 4.1b would be able to capture cross-terms

between A and B and would involve classification results from trouble sites created by merging A and B, which was not possible in Parks's approach. The Research Gap 2 highlights the requirement of efficient methods to learn the cross-terms. HDSM refines the process of creation of trouble sites by proposing an architecture consisting of a hierarchy of trouble sites, which are formed by merging the distributed sites in pairs of two. This would ultimately help in capturing cross-terms that may exist between the subset of sites, as shown in the above example, which was not possible to capture in Parks' methodology.

In order to maintain the performance of classification of vertically distributed data sources in a streaming context, another property of HDSM architecture suggested that the trouble sites may be added or removed from the hierarchy of trouble sites. The reason for continuous updating of this hierarchy is to handle the changes that may occur in the underlying patterns and cross-terms due to incoming data streams. Correspondingly in HDSM, three sets of monitors were used to provide an online and adaptive approach for forming the trouble site hierarchy structure: *creation*, *removal*, and *blacklist monitors*.

The system is initialized with a minimal structure consisting of only the primary sites, the aggregator, and three sets of monitors. Also, there is no trouble site at the initialization stage. As the name suggests, the *creation monitors* are responsible for creating the trouble sites. On account of this, they keep an eye on possible source sites and trigger the creation of a trouble site when the agreement between any set of source sites reaches a threshold value for a predefined amount of time. The agreement is calculated at a trouble site, which defines the probability that all the source sites declare a record as trouble. With creation monitors, *removal monitors* track how frequently a trouble site is used, when it is used, and what its accuracy measure is. If the accuracy of a trouble site drops below a specific threshold value for a predefined amount of time, it is then removed and added to the *blacklisted monitors* list. A trouble site should be

unblacklisted for its regeneration, which normally happens during *concept drifts*. At this stage, the blacklist monitors send an indication to creation monitors, allowing them to regenerate a trouble site when required.

A creation monitor is created for each combination of the available sites that can be utilized as a source site. Remember, not all sites can be used as source sites. A blacklisted site cannot be a source site until it is unblacklisted. Also, a site that is already a source site to some existing trouble site cannot be considered a source site candidate for creating a new trouble site, as there would be a repetitive transmission of the same features to many trouble sites. The critical issue in vertically distributed data sources is *to develop a robust method for pairing primary sites*. We have seen that when sites are merged in pairs of two offers an advantage over structures, which suggests a simpler trouble site connecting all primary sites. Now, a big question is ***how to choose the best subsets of primary sites for the pairing?***.

The HDSM architecture uses an *expensive brute force strategy* to determine the best pairs of primary sites for the first layer of trouble site at level one. This approach checks for all possible combinations and selects the optimal pair depending upon the level of agreement¹ between them. If the agreement between any two source sites exceeds a *threshold value*², in that case, the trouble site is generated by combining these two source sites. This brute force identification process for pairing primary sites is not scalable with the number of distributed sites in the network because the number of alternatives explored grows exponentially with the number of sites, due to which the structure may witness an explosion of monitors and may collapse. We will illustrate this problem in detail using diagrammatic representation.

Initially, the structure contains just the primary sites representing the distributed sites and the aggregator for producing the final classification output. The model is initialized

¹Agreement is calculated based on the records classified with a confidence lower than the threshold value.

²This threshold value is used for triggering creation monitors for creating trouble sites

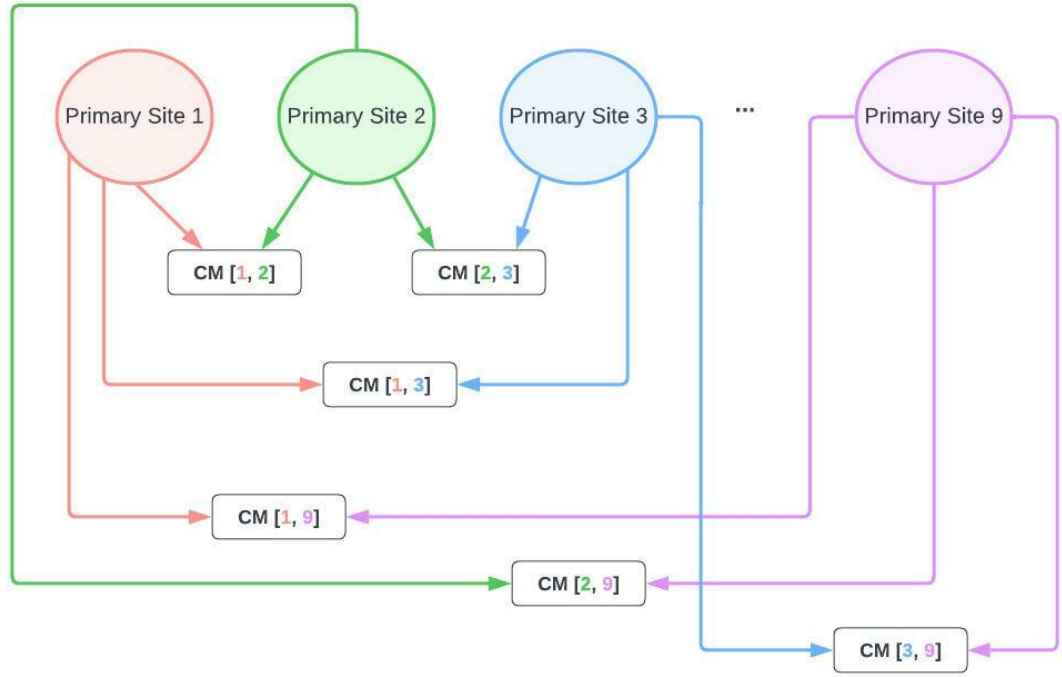


Figure 4.2: Initialization of creation monitors for a structure consisting of nine distributed sites.

with all the combinations of creation monitors. *Suppose*, there are nine primary sites ranging from 1 to 9; thus, the creation monitors would be initialized to $C(9, 2)$ different combinations, which comes to be 36, and these monitors will be responsible for holding the agreement information between the respective pairs as presented in Figure 4.2. Upon the arrival of every record, the creation monitors will update the agreement information and keep an eye on whether that agreement has reached the threshold value; if yes, a trouble site is created. In that case, the same procedure continues for the remaining sites as two primary sites have been used at this stage to form a trouble site. Thereafter, the creation monitors will observe the remaining seven sites following the same procedure and executing it in iterations. The overhead of regularly updating the agreement between all combinations and comparing it with the threshold value prior to creating trouble sites at the first layer is immense. As a result, we propose a novel method, *MTDM*, for learning the relationship between the distributed sites using a minimized tree instead of keeping overwhelming agreement calculations at the initialization stage when the

number of combinations checked is large. MTDM is targeted toward reducing the overload experienced on creation monitors for creating trouble sites on the first level of the structure.

4.3 Architecture

This section discusses the working of the proposed algorithm that instills the capability to learn the relationship between the distributed sites in a heterogeneous distributed data stream environment. However, the MTDM algorithm, to some extent, follows HDSM as the underlying structure, which proposes complete restriction on the transmission of records to the central site. Nevertheless, the MTDM algorithm suggests the transmission of a small subset of records from primary sites to the centralized site to generate a minimized centralized tree. As the transmission involves minimal communication compared to a centralized approach, this MTDM algorithm uses an approach that lies between centralized and decentralized mining.

After obtaining the minimized centralized tree, the tree is parsed to learn the relationship between the sites and filter out the most related sites. Following the basic ideology of trees, this centralized tree provides a list of features with maximum gain. As stated earlier, the system is initialized with primary sites, monitors, and the aggregator. The tree is generated at the initialization phase, which eases the process of creation of trouble sites. The role of a centralized site ceases once the pairs of sites revealing the best correlation have been learned. The fundamental idea used while parsing the tree is *to search for pairs with common trouble records greater than a threshold*³. The first step toward parsing the tree is to convert the features (or attributes) at the tree's nodes to their site, i.e., the site which contains that feature. Suppose, Site *A* contains features *F1* and *F2*; if any node in the generated tree contains *F1* or *F2*, then

³This is a standard value used in MTDM for filtering out the optimal pairs of distributed sites.

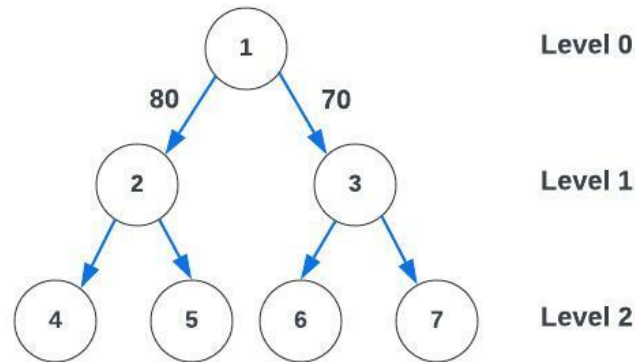


Figure 4.3: A sample centralized tree.

these features would be replaced by their site, A , in this case. The tree thus obtained is parsed breadth-wise to determine the coupling between the sites in terms of common trouble records to discover the best pairs of primary sites. *For instance*, if we obtained a centralized tree for some dataset as shown in Figure 4.3, nodes 1, 2, 3, 4, 5, 6, and 7 represent the features in the dataset. We can see that 80 records are going on the left side from the root depending upon the splitting value of the attribute (i.e., 1) present at the root, while the number of record flowing on the right side are 70. The algorithm maintains two lists: one for the left side and the other for the right side. The structure of the list would be a triplet of the form $\{(site\ 1, site\ 2)\ value\}$, where *site 1* and *site 2* are the pair of primary sites and the value represents the number of common trouble records between them. These lists, at any time, would provide details about the shortlisted pairs for their respective sides with a value representing the number of common trouble records between them.

We will now use an example to formulate the working of MTDM. Suppose we have three sites; $\{S1, S2, S3\}$ with the feature list $\{1, 2, 3, 4, 5, 6, 7\}$ split among sites as depicted in Figure 4.4 and Figure 4.3 represents the centralized tree structure for all these sites.

The procedure starts by converting the root feature, *node 1* in the above tree diagram

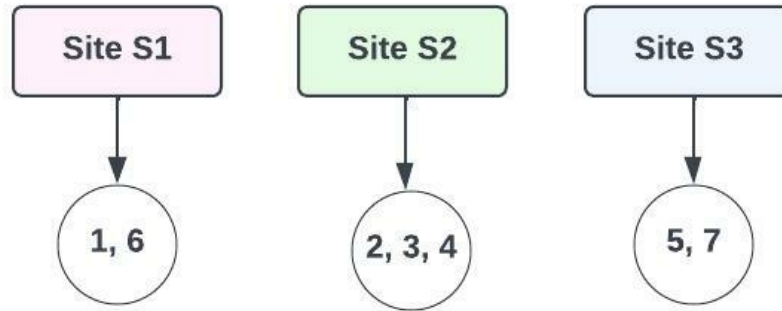


Figure 4.4: Vertically distributed sites structure.

in Figure 4.3, to the distributed site it belongs to, which is *S1* in Figure 4.4. The next step would be to scan the left child represented by *node 2* in the same tree, which eventually becomes *S2* because feature 2 is present at *S2*. Thereupon, the algorithm checks two conditions for processing the left child:

Condition 1: If the site present at root and left child is the same, then the algorithm will discard processing, and nothing happens for this child or else, the control jumps for checking *Condition 2*.

Condition 2: If (*S1*, *S2*) pair exists in the opposite side list, then the number of common trouble records would be added to the matched entry. Otherwise, the algorithm creates a new entry of (*S1*, *S2*) to the list of the side where *S2* lies (left or right) with the number of common trouble records between the sites.

After completing the scanning of the left child, the control moves to scan the right child denoted by *node 3* in Figure 4.3 by repeating the same procedure defined above.

In this example, on scanning the left and right child on the first level, the state of the tree would be the same as shown in Figure 4.5. For the left side consisting of pair (*S1*, *S2*), the if-part of both the conditions mentioned above; (*Condition 1* and *Condition 2*) becomes false. The reason for *Condition 1* being false is that *S1* and *S2* are two different sites. Furthermore, both lists are empty right now, so there is no possibility of finding

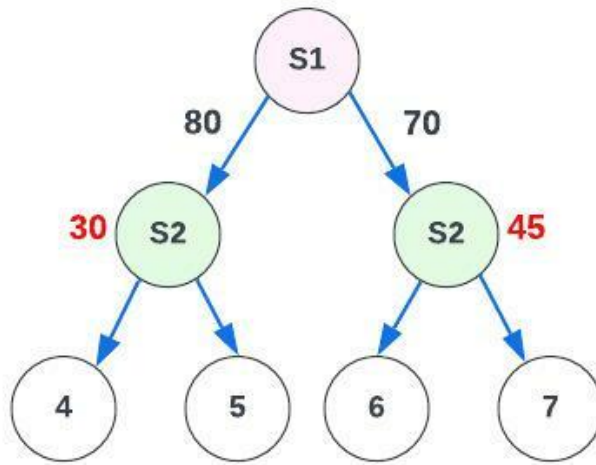


Figure 4.5: Pictorial representation of tree after parsing level 1.

the same entry on the right side list; therefore, *Condition 2* also became false. Hence, an entry of new pair $\{(S1, S2) x\}$ with x equal to common trouble records between $S1$ and $S2$ out of the total records flowing on the left side gets added to the left side list. The value of x , in this case, will be 30 (from Figure 4.5). Now, the algorithm control goes to the root's right child, *node 3* in Figure 4.3, and becomes $S2$ in Figure 4.5 after conversion to its originating site. The right side showed the same combination, that is $(S1, S2)$, making if-part of *Condition 2* true. Instead of making a new entry on the right side, the number of trouble records between $S1$ and $S2$ flowing on the right side gets added to the same entry on the left side. The left list would contain $\{(S1, S2) 75\}$ where 75 is obtained after adding 30 and 45, whereas the right side list would be empty. On moving down to level two, site $S2$ becomes the root, and *nodes 4 and 5* are scanned as the left and right child of $S2$ (parent). The left child (node 4) changes to site $S2$, turning *Condition 1* true, thereupon the algorithm discards processing the tree further in this direction. The parsing process continues and is similarly followed for the rest of the tree.

4.3.1 MTDM algorithm

The minimized tree is generated using the Hoeffding tree algorithm (Hulten et al., 2001). Hoeffding tree is a technique used for data stream classification that generates an approximate model using Hoeffding bounds (Hoeffding, 1963) with statistically guaranteed error bounds. Several parameters were configured for generating this Hoeffding tree, which were grace period, split confidence, and minimum records. The explanation and configuration of each of these parameters have been listed in Section 4.4.1. The minimized centralized tree was obtained by hyper tuning these parameters for generating a Hoeffding tree with the minimum number of records. The core functionality of MTDM lies in parsing this minimized tree. Also, the sensitivity of MTDM was evaluated depending on the values of these parameters in Section 4.4.4. Procedure `parsingtree` in Algorithm 7 is used to parse the tree. The minimized centralized tree generated by optimizing the parameters for the Hoeffding tree algorithm acts as input to the MTDM algorithm.

Algorithm 7 starts by initializing two empty lists, *trouble_records_left* and *trouble_records_right*, and a variable known as *maximum_trouble_records* is set to 0. The two sets of lists, as discussed above, are used for recording the pairs with common trouble records during the tree parsing process on both the left and right sides. While the variable reflects the highest value of common trouble records among these two lists. It would be worth mentioning here the tree obtained by applying the Hoeffding algorithm on any dataset is in the form of features or attributes seen at the distributed sites, i.e., the nodes of the tree represent the different features or attributes present at the sites. The scanning process starts by checking if there is a subtree in the tree as demonstrated in Figure 4.6; if yes, the attribute at the root is converted to the site which contains that feature. A subtree is a portion or subset of a larger tree structure that retains the hierarchical relationships and structure of the original tree. For this

algorithm, a subtree represents a collection consisting of a node (referred to as the root of the subtree) and its left and right children. For example, in Figure 4.6, nodes 2, 3, and 7 forms a subtree, with node 2 as the root and nodes 3 and 7 representing the left and right children of 1. Similarly, nodes 5, 6, and 7 form another subtree. The algorithm proceeds by checking if there is a left child of this root. Upon encountering the left child, the algorithm calculates the number of records flowing from the root to this child and checks the condition mentioned on line 9 in Algorithm 7. After ensuring that the number of records flowing from the root to the left child is greater than the variable *maximum_trouble_records*, the left child is also converted to the originating site. At this point, we have a pair with (*root site*, *left site*). Next, the procedure will check the conditions mentioned in Section 4.3 to see if this pair can be added to the lists (*trouble_records_left* or *trouble_records_right*). They can be two ways this pair can be added to the lists, either it can be added to *trouble_records_left* with the common trouble records between (*root site*, *left site*), if it represents a new pair on the left side, following the else part on line 16 in Algorithm 7 OR if this site pairing already exists on the list of the right side (i.e. *trouble_records_right*), then the common trouble records between *root site*, the *left site* is added to the common trouble records against that entry according to else-if part on line 13 in the same algorithm OR if *root site* is equal to the *left child site*, then the algorithm as mentioned on line 11 would discard this *left child site*. Upon completing processing the left child, the algorithm jumps to the right child, which is converted from attribute to site, subsequent to fulfilling the condition on line 9 of the same algorithm. The conditions mentioned in the list in Section 4.3 are checked in a similar fashion for (*root site*, *the right site*) to see if this pair can make their way to *trouble_records_right* OR can update *trouble_records_left* OR would nullify. Upon processing this subtree which involves parsing left and right children and updating the lists, the procedure *maxValue* in Algorithm 7 is called to browse through the two lists and update variable *maximum_trouble_records*. To illustrate it with a visual example,

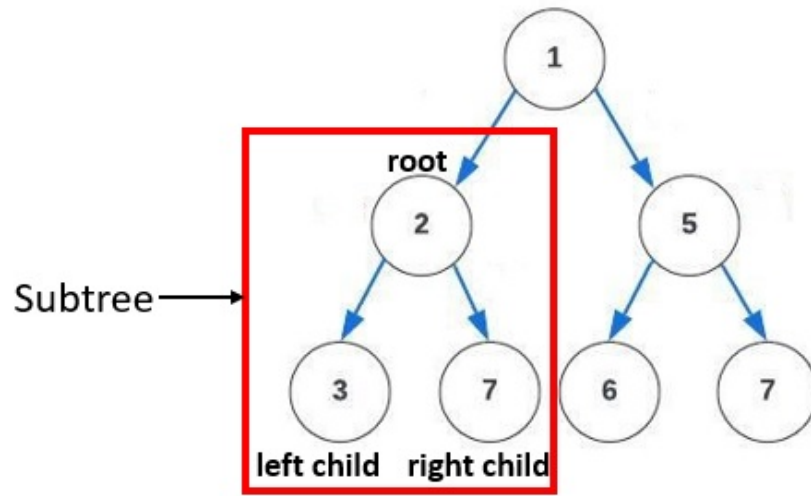


Figure 4.6: Graphic presentation of a subtree in a tree.

Figure 4.7a shows the state of a tree that has been scanned on the first level by the MTDM algorithm, where root and children sites are marked in different colors. Relative to this scanning process, Figure 4.7b also shows the structure of the list and processing of variable *maximum_trouble_records* upon completion of parsing the first level.

When the condition on line 5 in Algorithm 7 becomes false, and there is no subtree, the algorithm stops, and as output, it returns the optimal pairs of sites. For choosing the best optimal pairs through the MTDM algorithm, in the end, the algorithm scans the two lists and returns the pairs with common trouble records greater than a threshold. This threshold value is a user-configured parameter and can be tuned depending on the requirements. The tree is traversed breadth-wise, which offers an advantage of pre-stopping the parsing when the condition on line 9 checking "*if the maximum number of records flowing from root to child is greater than the maximum common trouble records seen so far using the variable maximum_trouble_records*" becomes false.

An example can demonstrate the process of pre-stopping; Figure 4.8a represents the modified version of a tree obtained for a site structure represented in Figure 4.4. Suppose 80 records are flowing on the left side and 70 are going on the right side,

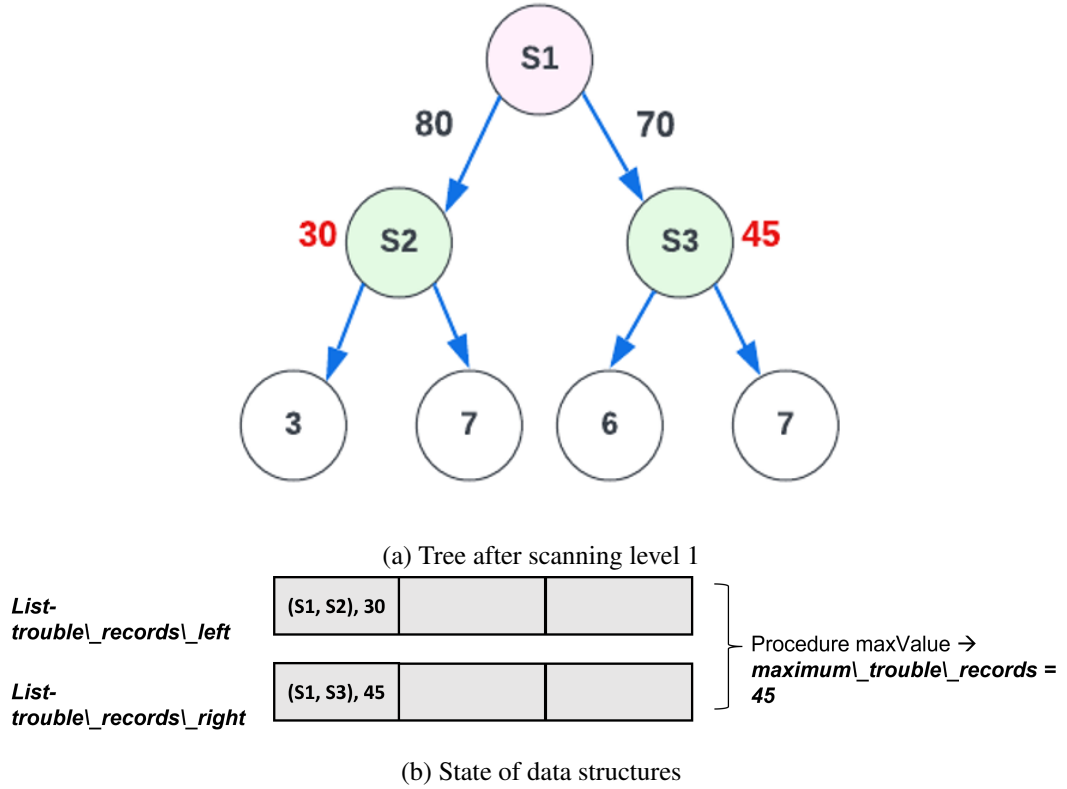


Figure 4.7: Demonstration of the tree, the two lists, and variable *maximum_trouble_records*, upon completion of parsing Level 1 of the original tree.

depending upon the value of the root attribute. Out of 80, 30 are common trouble records between sites S1 and S2, while on the right side, the number is 45 between sites S1 and S3. Thus, after traversing level first, *trouble_records_left*, *trouble_records_right* would consist of {(S1, S2) 30} and {(S1, S3) 45} respectively while the variable *maximum_trouble_records* holds the value 45 on completion of level 1 as presented in Figure 4.8b. On moving down to the next level, S2 becomes the root with nodes 3 and 7 representing the left and right child of S2. The algorithm checks the left child after converting it to the site, which is S2. As both root and child sites are the same, so no action would be taken after scanning the left child according to *Condition 1*. Whereas, on scanning the right child, shown as S3 on Level 2, the algorithm checks *Condition 1* and *Condition 2*, which are evaluated as False, thus representing a new pair (S2, S3).

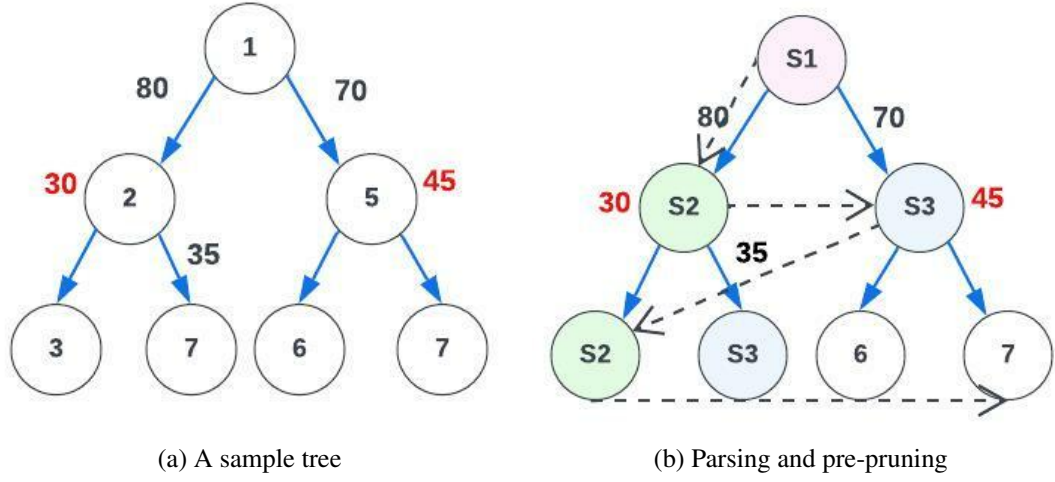


Figure 4.8: A) A sample tree obtained with nodes representing the features present at site, b) Depicting breath-wise parsing of tree represented in A) for pre-pruning.

The number of records flowing down in this direction, as shown in this example, is 35, depending upon the value of the splitting attribute. Now, the number of records seen on S3 is less than the value of *maximum_trouble_records*, so it stops scanning the tree further deep in this direction and declares it as the dead end. The reason for stopping here is that the common trouble records, in this case, will be trivial, and hence the method would continue its processing by exploring other neighbor nodes until it gets all dead ends or there is no subtree in the tree. As an outcome, the algorithm obtains the best primary site pairs, thus overcoming the overhead of checking all the combinations.

The working of MTDM has been summarised in a flowchart represented in Figure 4.9. After implementing this algorithm, the creation monitors are initialized to fewer combinations as it excludes the sites already selected by the tree as the optimal pairs, henceforth achieving efficient implementation and saving resources.

4.3.2 Complexity analysis

This section will draw comparability in terms of $\mathcal{O}(\cdot)$ between HDSM and MTDM for learning the relationship between the distributed sites in heterogeneous data sources

Algorithm 7: Procedures for finding the optimal pairs.

```

1 Procedure parsingTree (tree, dataset) :
2   Initialize maximum_trouble_records to zero
3   Initialize trouble_records_left list to empty
4   Initialize trouble_records_right list to empty
5   foreach subtree in tree do
6     start with root, convert the feature into site to which it belongs
7     foreach child of subtree do
8       calculate the number of records flowing in the child branch
9       if number of records flowing > maximum_trouble_records then
10        convert the child into the site to which it belongs
11        if site of child = site of root then
12          discard it
13        else if <root-site child-site> exists in the trouble records list of
14          other side branch then
15          calculate number of instances which the root site and
16          child site together declare as trouble
17          add the calculated value to the corresponding matching
18          value in the trouble records list of other side
19        else
20          calculate number of instances which the root site and
21          child site together declare as trouble
22          update the trouble records list to which the child belongs,
23          add the value calculated above to as a triplet (<root-site
24          child-site> value)
25        end
26      set the maximum_trouble_records to the maxVal
27      (trouble_records_left, trouble_records_right)
28    end
29  end
30
31 Procedure maxVal (list_a, list_b) :
32   list_x, list_y ← new list ;           // Initialize two lists
33   /* Copy only values in key-value pair from list_a
34      to list_x and from list_b to list_y */
35   foreach key-value pair (k, v) in list_a do
36     insert value v from triplet into list_x
37   end
38   foreach key-value pair (k, v) in list_b do
39     insert value v from triplet into list_y
40   end
41   max-val ← max (list_x, list_y)
42   return max-val

```

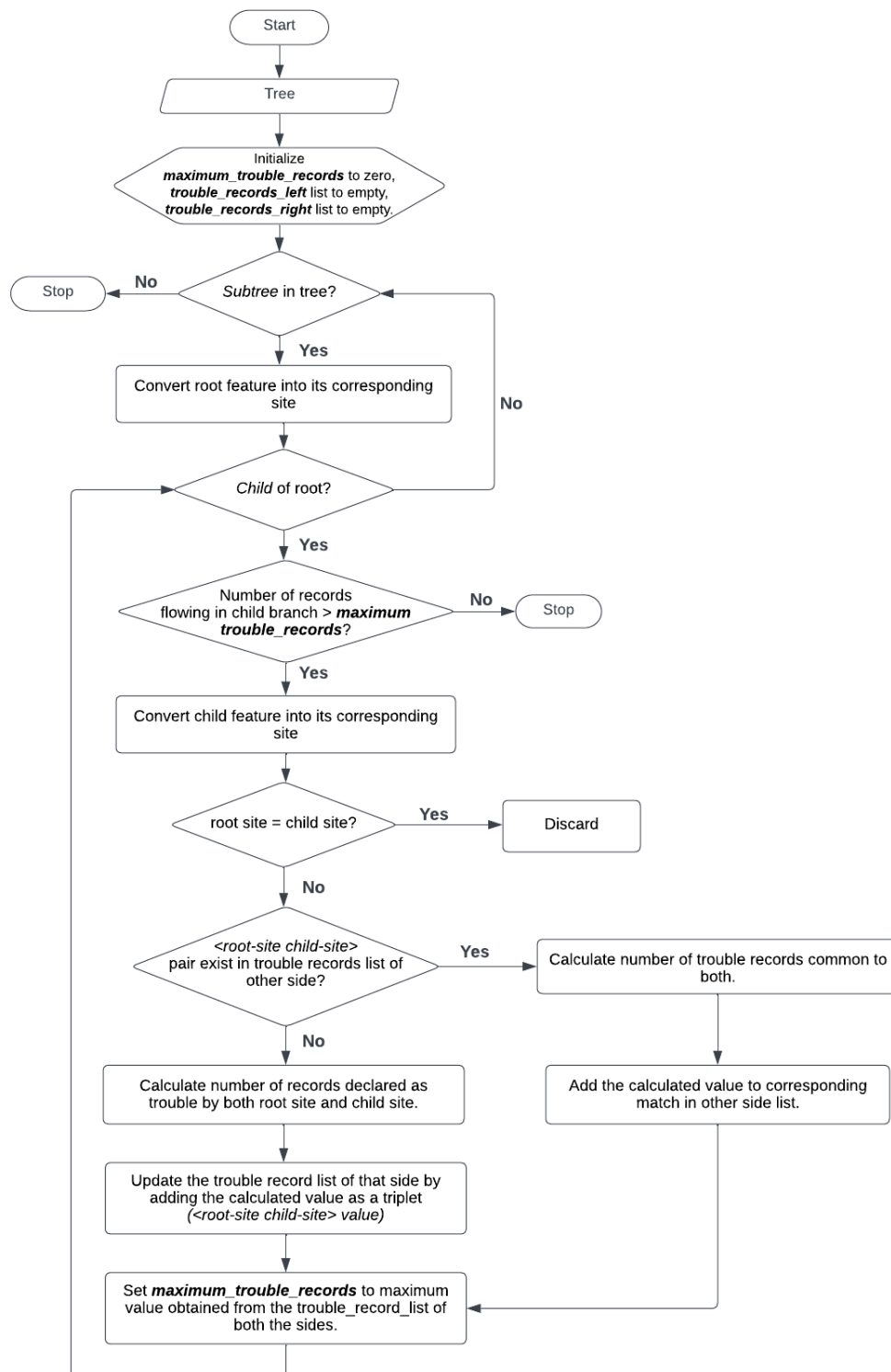


Figure 4.9: Pictorial representation of algorithm MTDM.

environment. In *HDSM*, as discussed before, it analyzes all the possible sequences to find the perfect union of sites. Therefore, it uses the $\binom{n}{k}$ formula for initializing its creation monitors, where n is the number of primary sites, and k represents the number of sites considered for pairing. Thus, the associated complexity would be $O(C(n, k))$ which can be written as $O(\frac{n!}{k!(n-k)!})$. On applying binomial expansion to this term, it comes out to be $((n-k+1)(n-k+2)(n-k+3) \dots n)/k!$. There can be one of two cases upon this either:

1. $n-k \geq k$
2. $n-k < k$

The complexity in the first case will be $O(n^k)$ and for the second condition, it will be $O(n^{(n-k)})$. Hence, the overall complexity turns out to be $O(n^{\min\{k, n-k\}})$, but in our case, k will always be 2 because we are pairing primary sites in pairs of two. Overall, if the number of sites in a heterogeneous distributed scenario is greater than 4, then the *Formula 1* would be used. Otherwise, *Formula 2* will resort. The number of combinations increases exponentially with the number of distributed sites. On top of it, when the agreement information requires updations for all the combinations on every record, it demands huge computational resources.

For *MTDM*, we propose generating a minimized centralized tree to overcome the brute force strategy for learning relationships between the distributed sites. As we are using datasets with only numeric attributes, an assumption can be made in this case that we will always obtain a binary tree [refer to MOA (Bifet et al., 2010) for more details]. However, the Hoeffding tree thus obtained can choose to split on each numeric attribute more than once; therefore, the number of attributes cannot determine the maximum size of the tree. The complexity of generating a Hoeffding tree is of the order $O(n.m)$, where n is the number of instances and m is the number of attributes. The complexity of parsing breadth-first search is $O(n + m)$, where n is the number of nodes and m is

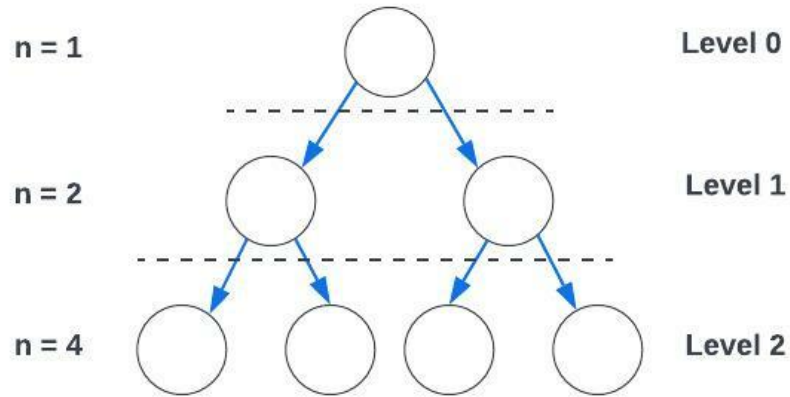


Figure 4.10: Complexity analysis of parsing the tree.

the number of edges. As we will obtain a binary tree, the complexity turns out to be $O(n)$. In Figure 4.10, the first layer contains two nodes; on scanning these nodes, the complexity turns out to be $O(2)$ on this level. Moving down to level two, if all the nodes on this layer are parsed, the total complexity becomes $O(7)$ (including the root node, as the root node also requires processing in this algorithm). MTDM uses only a small subset of records to build this tree; moreover, the algorithm does not get involved in scanning the entire tree. In fact, it ceases processing the branch where number of records becomes inconsiderable, as explained in the previous section. It should be noted that *the nature of the dataset* and *the number of vertical partitions* plays a vital role in analyzing the complexity. Therefore, by implementing MTDM, we add a slight complexity for generating and parsing the tree, which saves us from maintaining a vast number of creation monitors.

4.3.3 Aggregation methods

There can be various methods to agglomerate the findings from local models to produce the final result at the aggregator, such as *maximum confidence*, *majority voting*, and *stacking*, which have already been explained in Section 3.3.3.

4.4 Experimental Evaluation

The experiments for state-of-the-art work and MTDM has been made available in a GitHub repository; refer to Appendix C for details.

4.4.1 Experimental setup

MTDM algorithm has been implemented in the Clojure programming language with the help of machine learning libraries from the MOA data stream mining system (Bifet et al., 2010). All experiments were performed with OpenJDK 1.8.0 on a 64-bit Windows running on a 4×3.40 GHz Intel Core i5 CPU with 16 GB of memory. We have used ARF (H. M. Gomes et al., 2017) classifiers at trouble sites while Naïve Bayes classifiers were used at the primary sites. The default aggregation method used for generating the final classification was *maximum confidence*. There were several configuration parameters that were kept fixed throughout the implementation phase in HDSM (Denham et al., 2020b). They were *size-limit for all windows*, *all threshold values for creation and removal monitors*, *Hoeffding bound (δ)*, *smoothing factor for agreement threshold (θ)*, *and TF*, all these parameters have been referred to in Section 3.4.1. As this algorithm aims to come up with an optimal way to detect the best subsets of distributed sites for pairing, thus the reduction in the number of combinations will be the same for all three cases of TF. Therefore, we are reporting the results when TF is 1x. An additional set of parameters that were given constant values for executing MTDM were:

- Grace period
- Split confidence
- Minimum records

Grace period and *split confidence* are the parameters that are configured for the Hoeffding Tree generation algorithm. *Grace period* denotes the number of instances that must

be seen by a leaf node before splitting, while *split confidence* is one minus the desired probability that the correct attribute is chosen at the splitting point (Prasad & Agarwal, 2017). *Minimum records* are a subset of records transmitted to obtain a minimized tree. A threshold value that is used for filtering out the best pairs with maximum relationship is termed as *selection records*. This value is used for comparing, and pairs with common trouble records more than this threshold value are reported as optimal pairs.

The experiments were performed on eleven datasets, out of which nine datasets were real-world: NASA FLTz (FLT), EEG Eye State (EEG), Sensorless Drive Diagnosis (SDD), AutoUniv (AU), HIGGS (HIG), Fog Release (FOG), Wall-Following Robot Navigation (WFR), Power Usage (POW), Prosper Loans (PRL). Two were synthetic: RBF-F and RBF-M, based on the Radial Basis Function generator (RBF) and SEA generator. Table 4.1 outlines the properties of the dataset; the explanation of each property term has been provided in Section 3.4.1. FLT and PRL data were retrieved from NASA's data portal (N. Oza, 2011) and Prosper Marketplace (Prosper Marketplace, Inc., 2014), respectively. The rest of the real-world data were retrieved from UCI Machine Learning Repository (Dua & Graff, 2017), whereas the synthetic datasets, RBF-F and RBF-M, were generated with MOA (Bifet et al., 2010). The criteria used for selecting datasets for testing the performance of MTDM were based on deriving a scenario with a primary site count equal to or greater than nine and exhibiting cross-terms between distributed feature sets.

4.4.2 Performance assessment metrics

Several metrics have been used to analyze the algorithm's performance; Table 4.2 provides the summary for these metrics. An explanation for *accuracy* and *TT* has been discussed in Section 3.4.2. Similar to PODSM, the methodology used in MTDM is Prequential evaluation; that is, each instance is first utilized for testing the model

Table 4.1: Performance evaluation dataset properties.

Dataset	Feature count	Feature per site	P-site count	Class count	Record count	Stream
AU	53	1	53	2	10,000	Stream
POW	10	1	10	3	19,735	Stream
SEN	48	2	24	11	58,509	Interleaved
HIG	19	1-3	19	2	30,000	Stream
PRL	10	1	10	2	12,682	Stream
FOG	9	1	9	3	38,774	Shuffled
FLT	20	1-3	9	2	25,034	Shuffled
EYE	14	1	14	2	14,980	Shuffled
WFR	24	1	24	4	10,913	Stream
RBF-F	10	1	10	5	50,000	Fast incremental drift
RBF-M	10	1	10	5	50,000	Moderate incremental drift

Table 4.2: Explanation of metrics used for performance evaluation.

Metric	Explanation
Accuracy	Classification accuracy
TT	Sum of total data transmission to trouble sites and transmission for generating a tree
TC	Combinations checked to find the optimal pairs
TCM	Time taken to initialize creation monitors
TFTS	Time utilized prior to creation of first trouble site
TUCM	Time utilized for updating the creation monitors

before further being used for training. For MTDM, the TT metric includes two types of communication: one that covers the data transmission to trouble sites, and the second involves the communication for generating a minimized centralized tree. *Total Combinations (TC)* are the number of combinations to which the creation monitors are initialized. *Time for Creation Monitors (TCM)* is the time taken to initialize the creation monitors, while *Time for First Trouble Site (TFTS)* accounts for the total time taken to hunt for optimal pairs for forming the trouble sites at layer one. *Time for Updating Creation Monitors (TUCM)* represents the total time utilized for updating the creation monitors.

4.4.3 Performance comparison

The major aim behind MTDM is to propose a scalable and efficient method for learning relationships between distributed sites. The following section will report the result of applying MTDM on heterogeneously distributed data streams.

Table 4.3 illustrates the accuracy comparison between MTDM and state-of-the-art approaches, with bold values representing the best value. We generate an accuracy comparison between a *distributed ensemble using maximum confidence*, *Parks' algorithm*, *HDSM*, and *MTDM*. It can be noticed that MTDM has the potential to achieve improved classification accuracy over other methods for the majority of the datasets. Another comparison has been drawn between HDSM and MTDM using metrics; TT, TC, TCM, and TFTS in Tables 4.4 and 4.6 to illustrate the efficiency of MTDM in terms of resource time for finding the optimal pairs of distributed sites. The reason for choosing HDSM for this comparative analysis is that only HDSM and MTDM methodologies suggest pairing of distributed sites in subsets of two; hence HDSM would be the most comparative work to MTDM. It can be seen that the number of alternatives (TC) explored for forming trouble sites decreases significantly for all the datasets upon execution of MTDM. In addition to reducing the number of combinations examined, the algorithm proves its potential in terms of resource time, TCM, which is expressed in milliseconds (ms). Table 4.4 shows a decline in the time required for initializing creation monitors for all the datasets after implementing MTDM compared to the brute force strategy used in HDSM. Time taken by a dataset to find the optimal pairs is affected by the dataset's size and nature, the number of vertical partitions present in the dataset, and the relationship seen among the partitions.

For MTDM, communication increases because the TT metric includes two types of communication cost as stated earlier; *firstly*, the transmission that occurs due to trouble records and *secondly*, a small amount of transmission that occurs to create a minimized

Table 4.3: Comparison in terms of accuracy.

Dataset	Distributed Ensemble	Parks' method	HDSM	MTDM
POW	46.01	46.03	58.99	57.87
SEN	53.52	53.52	79.10	79.46
HIG	53.19	53.19	53.59	53.29
PRL	63.43	63.38	64.19	64.31
FOG	68.22	68.22	84.41	85.53
FLT	67.67	67.67	79.00	78.17
AU	64.27	64.27	62.26	62.34
EYE	48.98	48.98	50.06	50.39
WFR	58.48	58.48	79.18	79.24
RBF-F	30.20	30.20	34.82	34.12
RBF-M	32.68	32.68	43.92	43.92

central tree. The transmission of records for generating a central combiner tree puts a small transmission cost over HDSM, which can be seen in Tables 4.4 as TT. In contrast, this cost is much less when compared to the centralized structure of forwarding all the records to the centralized site. Table 4.5 draws the difference between the centralized setup and MTDM. It can be seen that though MTDM adds a small communication cost, this cost is significantly lower than the communication cost of centralized experiments except for one dataset, *SDD*. In this dataset, the increase in communication has been immense; at the same time, the number of combinations has also shown a sharp decline. In this case, the significant increase in TT can be attributed to the dataset's inability to generate a Hoeffding Tree with a minimum set of records, which may be due to reasons like complex relationships between features, lack of clear splitting criteria, etc. Overall, a reduction in the number of combinations checked generally comes at the expense of smaller communication costs for all the other datasets.

Another metric used for comparing both methods is *TFTS* (reported in milliseconds), which defines the time consumed until the first trouble site is added to the architecture. As mentioned, HDSM initializes creation monitors for all the combinations of primary sites. Upon the arrival of every record, it updates the agreement and maintains a check

Table 4.4: Comparison in terms of communication, combinations, and TCM.

Dataset	HDSM			MTDM		
	TT	TC	TCM	TT	TC	TCM
POW	460.75	45	2.749	661.03	23	0.588
SEN	555.70	276	2.427	2620.02	193	1.880
HIG	563.90	36	1.276	556.01	10	0.475
PRL	361.76	45	2.747	493.38	29	0.558
FOG	328.62	36	1.679	366.61	22	0.529
FLT	430.53	36	1.930	559.80	17	0.486
AU	998.32	1378	6.241	2083.60	1276	2.891
EYE	400.21	91	1.669	597.42	67	1.256
WFR	746.23	276	2.564	1454.08	232	1.678
RBF-F	465.55	45	2.432	670.04	23	0.766
RBF-M	483.31	45	2.445	683.31	29	0.839

Table 4.5: Comparison between centralized mining and MTDM.

Dataset	Centralized Experiment		MTDM	
	Accuracy	TT	Accuracy	TT
POW	69.59	1000	57.87	661.03
SEN	90.98	2400	79.46	2620.02
HIG	51.90	900	53.29	556.01
PRL	65.65	1000	64.31	493.38
FOG	91.09	900	85.53	366.61
FLT	84.94	900	78.17	559.80
AU	64.20	5300	62.34	2083.60
EYE	79.99	1400	50.39	597.42
WFR	92.16	2400	79.24	1454.08
RBF-F	67.34	1000	34.12	670.04
RBF-M	79.67	1000	43.92	683.31

to ensure if any combination's agreement has reached a threshold. This checking for a large number of combinations is repeated until the first trouble site is created because after that point the number of combinations scanned starts decreasing. However, after applying MTDM, a fraction of the time is consumed in generating the Hoeffding Tree and parsing it to obtain the optimal pairs in terms of common trouble records which is considerably low than the overhead involved in HDSM. Therefore, it allows initializing

Table 4.6: Resource time comparison in terms of TFTS.

Dataset	HDSM	MTDM
POW	3332.96	1535.30
SEN	20923.78	14784.16
HIG	3353.04	1523.94
PRL	3362.75	2270.14
FOG	2653.76	1654.54
FLT	2945.13	1387.84
AU	96360.23	81260.56
EYE	6159.65	4633.25
WFR	17410.96	16112.27
RBF-F	3412.36	1667.83
RBF-M	3206.46	2075.70

Table 4.7: Comparison for other aggregation methods.

Metrics	HDSM			MTDM		
	MaxConf	Voting	Stacking	MaxConf	Voting	Stacking
Acc	79.00	80.79	80.22	78.17	81.75	81.25
TT	430.53	528.08	448.51	559.80	642.18	574.59
TC	36	36	36	17	17	17
TFTS	2945.13	2631.23	2682.01	1387.84	1199.55	1250.11

the creation monitors with fewer combinations, reducing time and computational overhead in regularly updating and comparing them with threshold values. Table 4.6 reveals a gain in efficiency in terms of computational resource time (expressed as TFTS here).

Table 4.7 presents a comparative study among all aggregation methods discussed in the previous section; this comparison has been drawn in terms of all the metrics for the *FLT* dataset. It can be seen that though accuracy under the maximum confidence aggregation method witnessed a minor decrease, for other aggregation methods, there has been a slight increase in accuracy level. TC observes the same value for HDSM and MTDM as shown in Table 4.7 under all three aggregation methods.

A collation analysis depicting the implementation distinction between HDSM and MTDM has been presented in Figures 4.11 and 4.12, which shows the timelines for

various metrics for the *FLT* dataset. The *first graph* in both of these figures shows the accuracy level changes during the experiment's execution and depicts the cycle of creation and removal of trouble sites. The *red color horizontal plot* depicts the accuracy levels while the *green and red color vertical lines* portray the continuous process of addition and removal of trouble sites, respectively. The *second graph* reports the data transmission volume as discussed in Section 3.4.3. The *last graph* in Figures 4.11 and 4.12 outlines the time utilized to maintain the creation monitors. This includes the time taken to update the agreement of all creation monitors and regularly keeping a check by comparing the agreement factor with the threshold value. On examining the execution of both experiments, we can find that the accuracy statistics maintain an analogy between both implementations. At the same time, a marginal improvement has been seen in the transmission factor during the running phase of the experiment. However, a small communication which is required at the initialization phase for creating a tree is not shown in Figure 4.12. A significant gain in terms of resource time can be spotted from this comparative analysis. The resource time in both the figures reaches their maximum value at a record *1400*, after which a set of trouble sites is created. A steep fall in resource time utilization can be seen in Figure 4.12 with the application of MTDM, as the number of combinations explored by the creation monitor decreases. Though the trend is the same for both curves, there is a large difference in values, thereby inducing huge savings in resource time.

A comparative study has been presented between HDSM and MTDM concerning *relative ranks in accuracy, communication, and resource time*⁴ in Figures 4.13 and 4.14. For relative ranking between accuracy and TUCM, the graph in Figure 4.14 shows the competency of MTDM in achieving a substantial improvement in resource time. Moreover, it also demonstrates the algorithm's capability in maintaining a better trade-off between *accuracy and resource time* as it lies near the origin point compared

⁴It is expressed as TUCM accounting for the total time taken for updating creation monitors

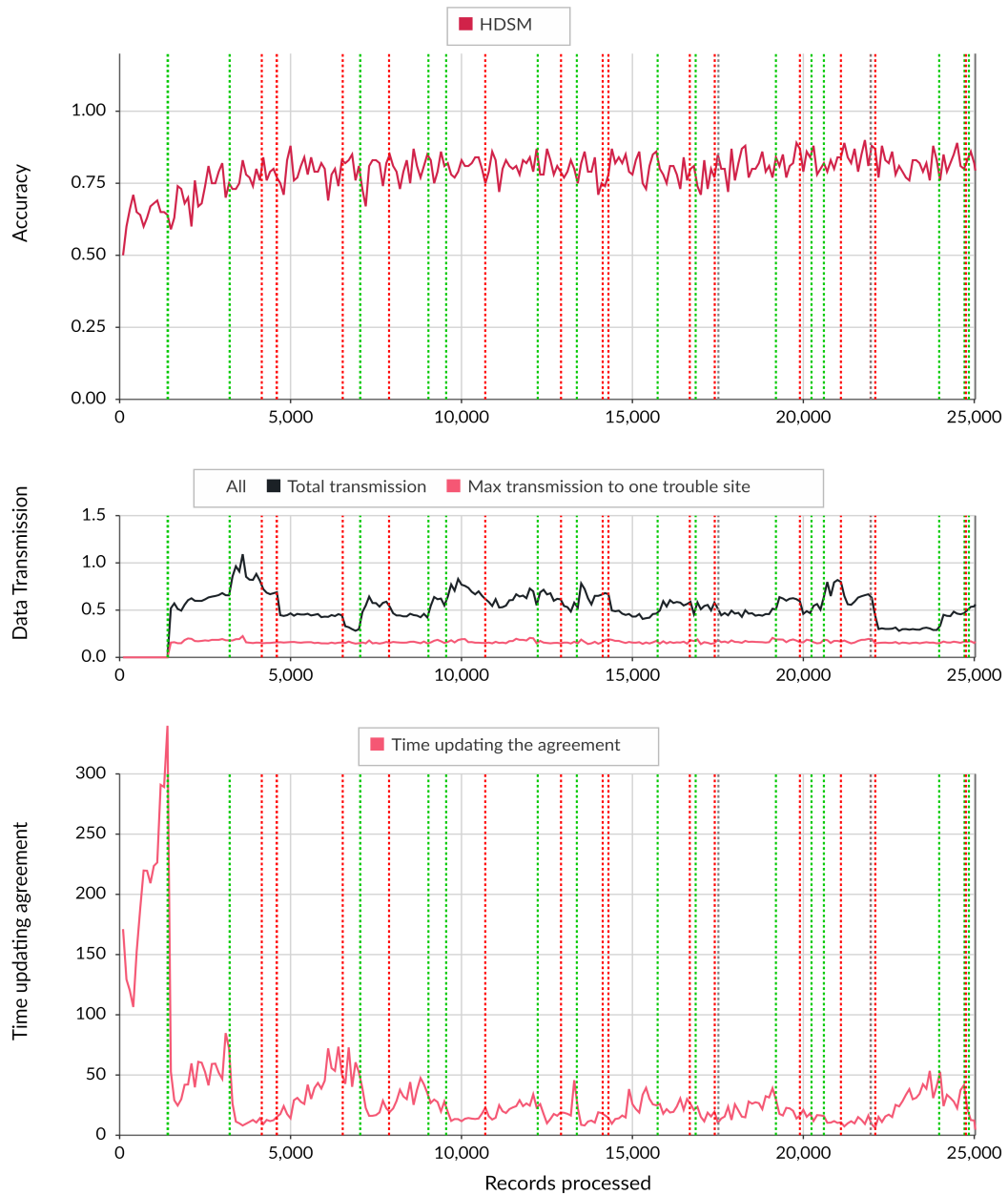


Figure 4.11: Timelines of accuracy, data transmission, and resource time with HDSM.

to HDSM. For ranks regarding accuracy and communication, we could witness a slight difference and the reason being small amount of communication consumed for generating the tree.

Finally, a comparative observation has been drawn in Figure 4.15 among the three different methods *a) centralized processing*, *b) HDSM* *c) MTDM* for three datasets with

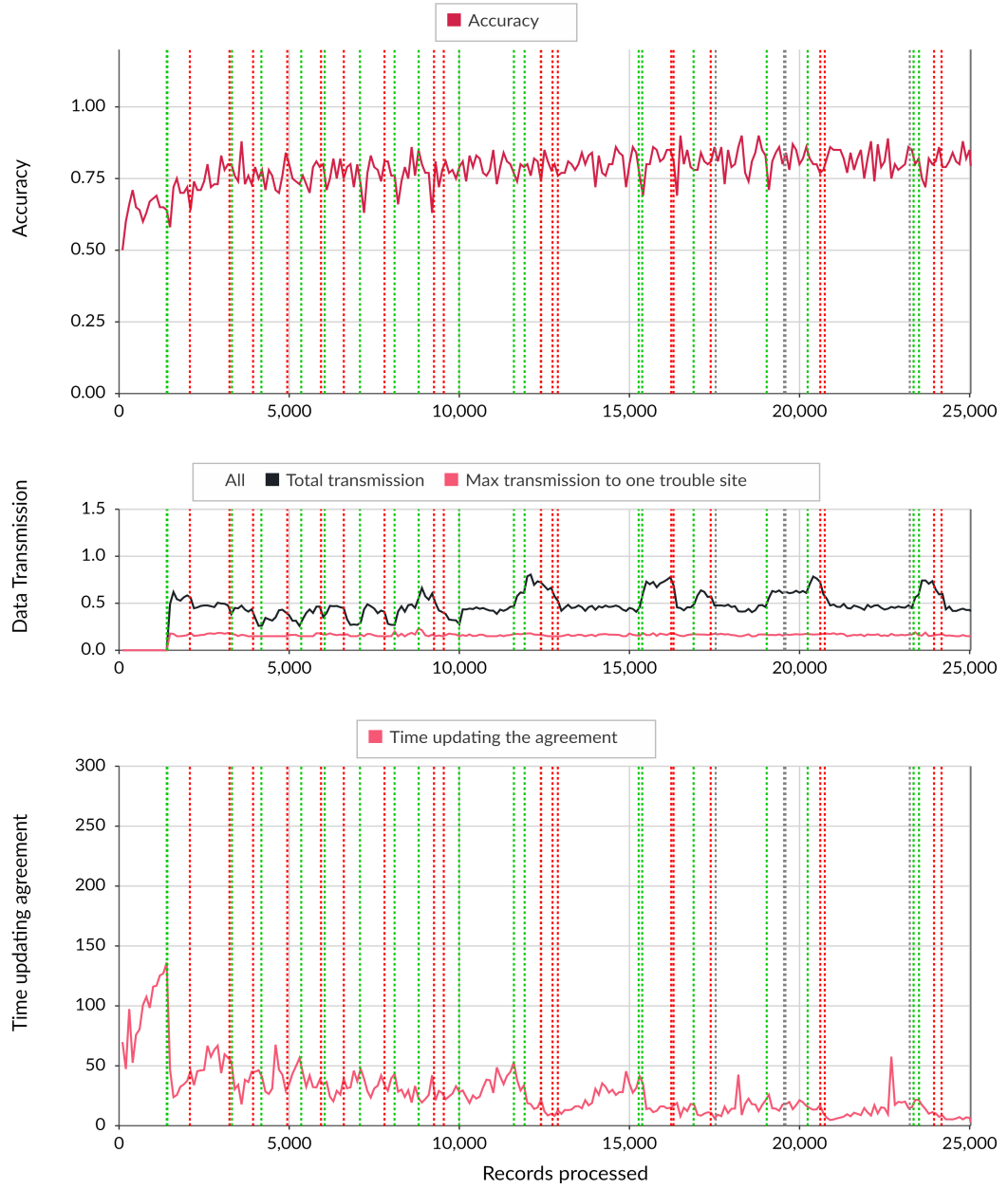


Figure 4.12: Timelines of accuracy, data transmission, and resource time after implementing MTDM.

different properties: *FOG* (with minimum number of primary sites), *AU* (with maximum number of primary sites), *SEN* (involving high communication for generating tree). For all the datasets, it can be observed that accuracy obtains its highest values for the centralized experiment, which is achieved at the expense of huge transmission. MTDM

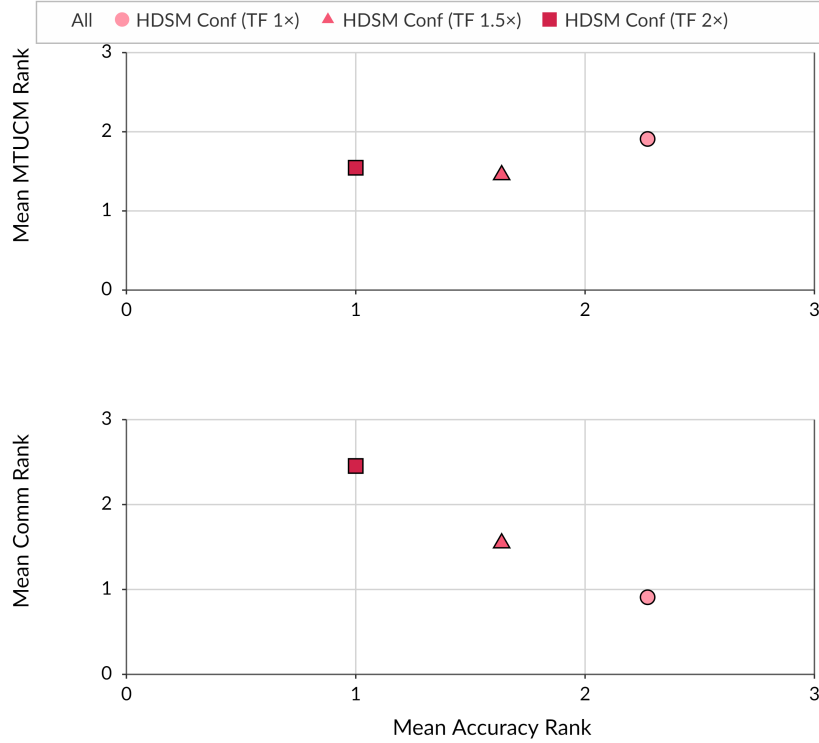


Figure 4.13: Relative ranks in HDSM.

adds minor transmission except for *SDD* (which includes an immense transmission) while improving accuracy for *FOG* and maintaining a similar accuracy level for *AU*. The results reported in Tables 4.4 and 4.6 illustrate the proficiency of MTDM in improving the performance of classifying vertically distributed data streams. The algorithm's ability is demonstrated to significantly reduce resource time and improve accuracy in most datasets with a minor increase in communication.

4.4.4 Parameter sensitivity analysis of MTDM

MTDM's sensitivity is evaluated based on the mutable parameters used in this study; the list of these parameters is:

1. Grace period
2. Split confidence

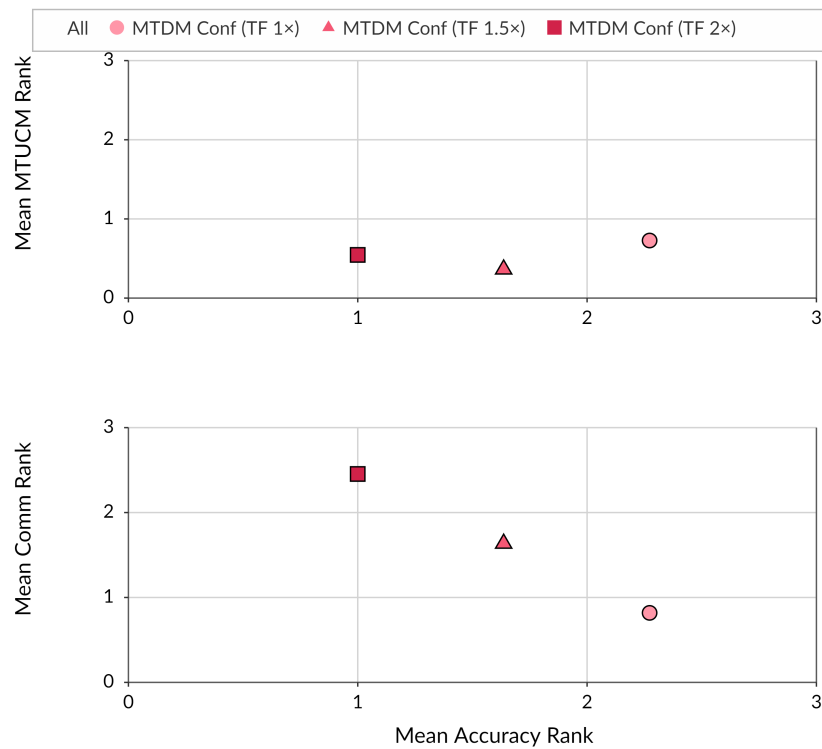


Figure 4.14: Relative ranking in MTDM.

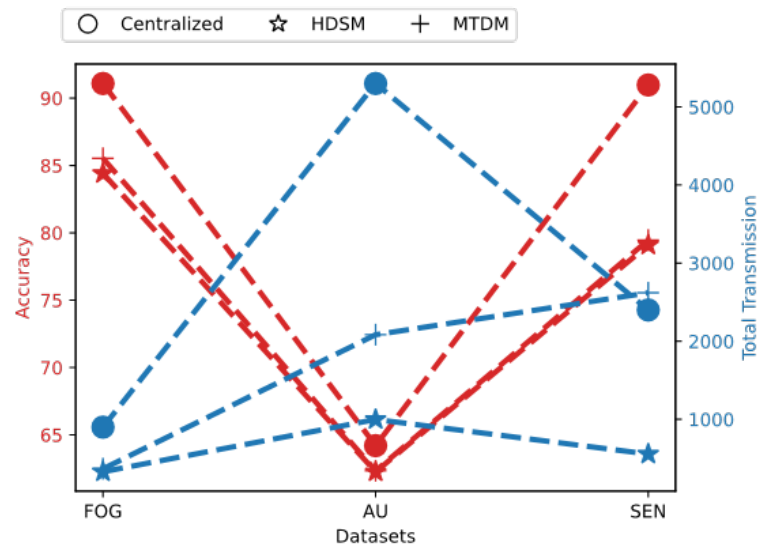


Figure 4.15: Differentiation between centralized experiment, HDSM, and MTDM.

3. Minimum number of records for generating central tree.

As stated above, *grace period* and *split confidence* are the parameters configured to

Table 4.8: Impact of changing values of sensitive parameters.

Parameters	Value	Acc	TT	TC
Grace period	10			
Split conf	0.0001	78.17	559.80	17
Min records	3000			
Grace period	5			
Split conf	0.0001	78.86	533.20	22
Min records	3000			
Grace period	10			
Split conf	0.001	77.53	574.45	13
Min records	3000			
Grace period	10			
Split conf	0.0001	78.17	631.70	17
Min records	5000			
Grace period	5			
Split conf	0.0001	78.63	621.55	13
Min records	5000			

obtain a Hoeffding Tree, while the *third parameter* defines the least number of records that are transmitted by each primary site to generate a minimized central tree. As the value of split confidence approaches zero, a longer time is required for a splitting decision. Table 4.8 presents the effect of modifying these parameters on the *FLT* dataset. It can be seen that updating the values held by these parameters impacts accuracy, communication, and the count of combinations.

The diagrammatic representation of updating the value of minimum records utilized for the generation of the tree has been depicted in Figure 4.16. It can be seen that the number of combinations decreases to a minimum value when 5000 records are used for generating a tree. Furthermore, no impact of increasing the value of minimum records can be seen at 7000 on the number of combinations. Resource time here reports the time taken for generating and parsing the tree. This figure clearly illustrates an obvious conclusion: on growing the number of records, the time taken to generate and parse the tree would increase, but this increment is minute compared to the load of maintaining

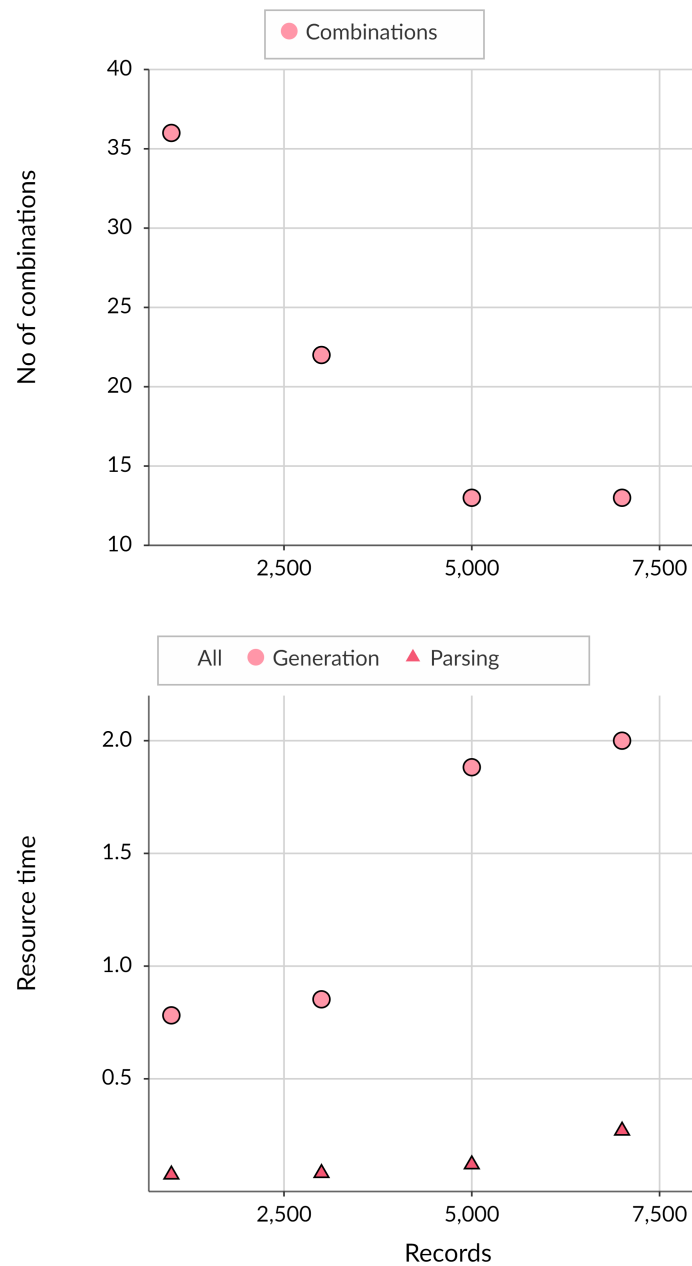


Figure 4.16: Impact of changing the value of minimum records.

the massive number of creation monitors. Therefore, the value of minimum records can be adjusted considering the resource constraints to maintain a balance between all the metrics.

4.5 Applicability and Limitations of MTDM

The experimental evaluation demonstrated the potential of the algorithm to select the optimal pairs of vertical partitions with lesser usage of resource time. It shows that the algorithm can be used for all cases: one with a smaller number of primary sites (*FLT and FOG with nine sites*) or in cases where a large number of vertical sites exist (*AU with 53 sites*). MTDM is a flexible algorithm that can be used with many datasets and other classification methods. It may even be applied to other machine learning scenarios, such as numeric regression. The algorithm will find its principal place in time-critical applications.

The fundamental limitation of MTDM, as discussed above, is when the communication overload becomes excessive, *specifically* when it becomes even more than the centralized structure, as seen in the SDD dataset. In this case, the relationship within the dataset is complex and requires greater data centralization. Thus, the solution to obtain the optimal pairs of vertically distributed sites comes at a considerable cost. The extreme case can be when a central tree generation requires complete transmission of all records from all the primary sites. For these cases, a mechanism could be developed to approximate a central tree without the transmission of records.

4.6 Chapter Summary

The focus of this chapter was on devising a method to learn the relationships between the vertically distributed sites for choosing the optimal pairs of primary sites for forming the trouble sites. To develop a scalable and efficient method, we proposed generating a central minimized tree using the minimal number of records to capture the relationships between the sites. However, the generation of this tree added a minor communication cost. At the same time, this robust technique saved the distributed mining process from

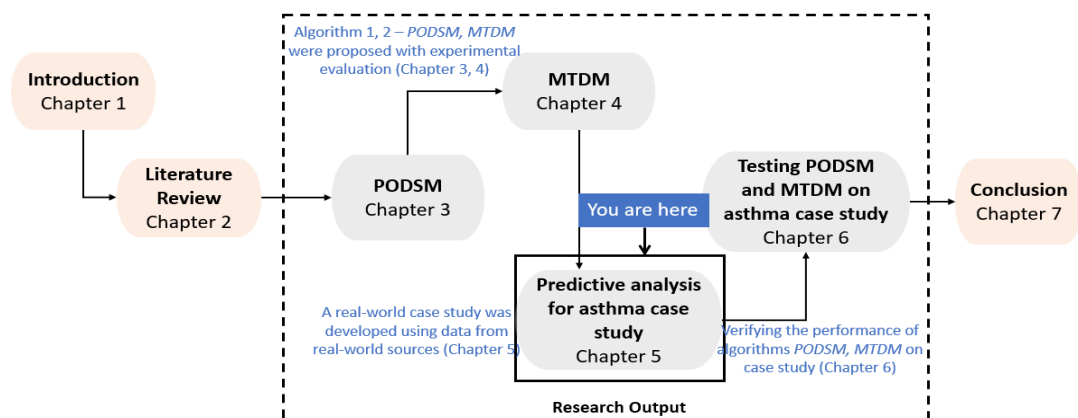
the risk of collapsing due to huge resource time consumption in the exploration of exponentially increasing combinations for pairing primary sites. MTDM proved to be an economical way of finding the optimal pairs of sites in the form of cross-terms.

Our results presented a significant reduction in the number of alternatives explored compared to the existing brute force strategy. The results show an accuracy improvement for most of the data sets while reducing resource time to nearly half. Although minor communication is used in MTDM for generating a tree.

Chapter 5

Case Study: Predicting Peak Demand Days for Asthma-Related ED Visits and Hospitalizations

This chapter is centered on building a live case study for validating the applicability of the algorithms, PODSM and MTDM. A deep reflection has been provided for each phase involved in generating the real dataset, which started with data collection from heterogeneous sources. Several machine learning algorithms were applied to build a prediction model by centralizing complete data from all sources in an offline context.



Roadmap of the thesis.

5.1 Introduction

Useful data generated through monitoring population health encompassing demographic and health surveillance and epidemiological studies on disease surveillance can play a prime role in health forecasting modeling (Soyiri & Reidpath, 2012). Health forecasting finds its application in providing early warning signals to health service providers in adverse events. *For example*, in the case of asthma, timely and accurate prediction of acute asthma exacerbations could facilitate medication management and other prevention strategies to prevent asthma worsening and the need for hospitalization due to a disease episode. The objective, then, is targeting preventive and therapeutic interventions. Health forecasting has been commonly used for decision-making, particularly for ED visits, daily hospital attendances, and admissions (Soyiri & Reidpath, 2012). The relevance of a health forecasting model can be realized in any of the following ways:

- Strengthening preventive healthcare services
- Issuing warnings for managing adverse events like overflowing of patients
- Cutting down costs associated with supplies and staff redundancy

Asthma affects more than 339 million people worldwide (*The Global Asthma Report 2018*) and is more common in developed countries like the U.S., as per a 2019 study published by Enilari and Sinha (2019). NZ is one of the countries with the highest prevalence and mortality rates due to asthma. OECD statistics indicate NZ has one of the highest hospital admission rates for asthma of OECD countries. This practical challenge motivates us to apply our research in the form of proposed algorithms, to find potential solutions for dealing with the asthma problem from a data perspective rather than clinical perspective. We are working towards developing prediction modeling for anticipating asthma-related hospitalizations in Auckland, NZ, while exploiting diverse heterogeneous data sources. These sources include: data for *meteorological factors*,

air quality, and Google Trends. This chapter thoroughly examines the phases involved in developing a prediction model. A comparative examination has been conducted based on the prediction models developed using different machine learning techniques, including ensemble modeling on the continuous data of 1097 days. The subsequent initiative was to build a balanced classification problem by generating synthetic data for the minority class using techniques like *SMOTE* and *random over-sampling*.

5.2 Problem Definition

Asthma accounts for more than 400,000 hospitalizations annually in the U.S., and these inpatient stays often come with physical and mental side effects (Bradley, 2022). NZ also experiences a high prevalence of asthma, with one in seven children (13 percent) aged 2–14 years (110,000 children) and one in eight adults (12 percent, 452,000 adults) taking current asthma medication. Respiratory disease is NZ's third most common cause of death. According to Asthma Respiratory Foundation NZ (2019), on average, 77 Kiwis die from asthma each year, and the death rate has not improved over the last few years. In the year 2015, there were 87 deaths due to asthma, although the rates calculated over seven years for 2007-2015 were lower than in the preceding seven years. However, the years 2014 and 2015 showed sudden spikes compared to the tenure of the previous seven years, as demonstrated in Figure 5.1.

OECD statistics indicate NZ has one of the highest hospital admission rates for asthma among OECD countries. A slight decline was witnessed in such hospitalizations over eighteen years, from 2000-2017. The asthma hospitalization rates have continued to acquire a value below 170 per 100,000 people each year over the last five years. The value reaches its lowest point of 157 in the year 2016. Indeed, the average of asthma hospitalizations from 2000-2012 were 198 and never observed a value below 180, as

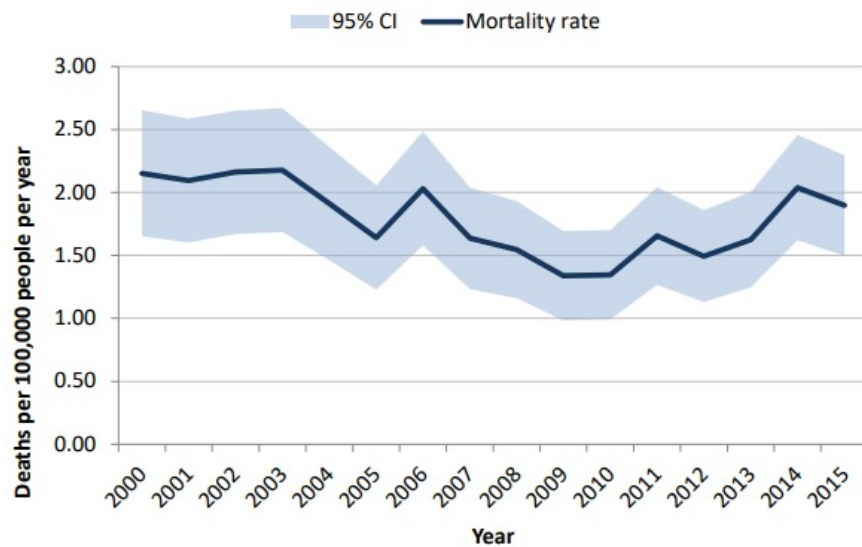


Figure 5.1: Asthma mortality rate from 2000-2015 with age-adjusted by Asthma Respiratory Foundation NZ (2019).

shown in Figure 5.2. The *cost*¹ of asthma to the NZ economy was estimated at \$349m in 2000, \$800m in the 2014 report, and at \$858m in the 2016 report (Barnard & Zhang, 2019). In 2015, asthma cost \$1.018b with \$198.3m in direct costs and \$828.2 in indirect costs from workdays lost, disability-affected life years, and mortality. Asthma incurred approximately \$121,993,745 cost in the year 2015 for ED and outpatient visits. The main cost of asthma is from exacerbations, but at present, it is very difficult to predict these, meaning that most patients and clinicians do not know when an attack might occur until it is too late. As such, there is a crucial opportunity for predictive modeling to help predict events before they occur to allow early medical intervention. Although asthma cannot be cured, there can be methods to minimize the impact of asthma on outcomes and economic costs, by targeting exacerbation and hospitalizations, which are the critical contributors to morbidity, mortality, and costs.

As discussed in Chapter 2.5, the predictive models developed for asthma can be broadly categorized into two depending upon the modeling type (*personalized or generalized*). A plethora of work has been done towards developing a prediction

¹All costs reported here are in New Zealand Dollars.

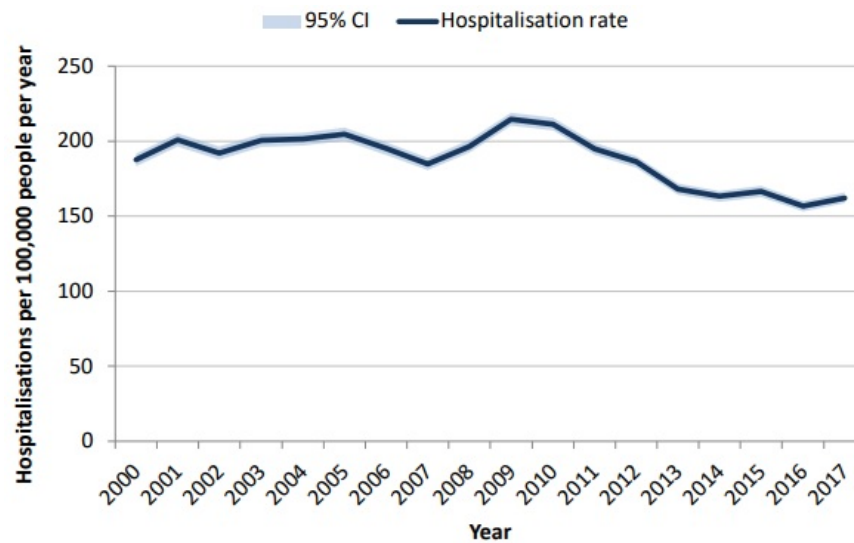


Figure 5.2: Asthma hospitalizations per 100,000 people per year for the period of 2000-2017 by Barnard and Zhang (2019).

model and was discussed in Sections 2.5.1 and 2.5.2. Most of these studies focused on *individual-based (personalized)* models for asthma whereas in this study, we concentrate on *population-based (generalized)* predictive modeling for asthma. There is less work undertaken towards a population-based model for asthma, and it has not been well examined, specifically in NZ.

The predictive modeling for asthma encompassing communities at different locations may not observe similar behavior and instead, these studies may produce uncertain outcomes due to a variety of reasons:

1. The concentration of pollutants may witness the differing magnitude of correlation with the meteorological factors during the year.
2. Different location experiences varying climatic conditions due to their geographic locations.
3. The mechanism used for measuring different parameters attributable to pollutants or meteorological factors differs from place to place, such as the number of monitoring sites, the frequency of recording data, and the particle size.

4. Respiratory conditions like asthma and COPD may affect different proportions of the population that may differ from one country to another, and also, the structure of populations may be dissimilar.

The geographic area under study was *Auckland*. According to our knowledge, there has not been any study centering on a prediction model that could predict peak demand days for asthma ED visits and hospitalizations for the indicated area. Several factors trigger and exacerbate asthma: weather conditions and environmental factors. Generating a model leveraging diverse data sources to predict asthma-related ED visits and admissions at a population level, would provide timely and targeted influences to minimize the socio-economic impact of asthma. In this chapter, we aim to exploit the distributed data sources to study the relationship between heterogeneous parameters and asthma-related hospitalizations, thereby using them to generate a prediction model.

5.3 Stages in Predictive Modeling

Developing a predictive model consists of several stages, starting from acquiring data from diverse sources to evaluating the final model. The diverse data sources examined in this study include *meteorological parameters, air quality parameters, and search trends*. Factors under each category are analyzed for their relationship with asthma-related visits and admissions. Finally, a refined list of predictors is utilized to create a prediction model, pictorial representation of this prediction model has been rendered in Figure 5.3. Furthermore, a relative analysis has been generated comparing the performance of prediction models developed, using different machine learning techniques, *ANN, RF, SVM, XGB, and Ensemble Classifier*. Work has also been conducted for oversampling techniques like *SMOTE* (N. Chawla et al., 2002) and *random over-sampling* (Kumar, 2020), aspiring to procure a balanced classification problem. *Below is the list of all the stages:*

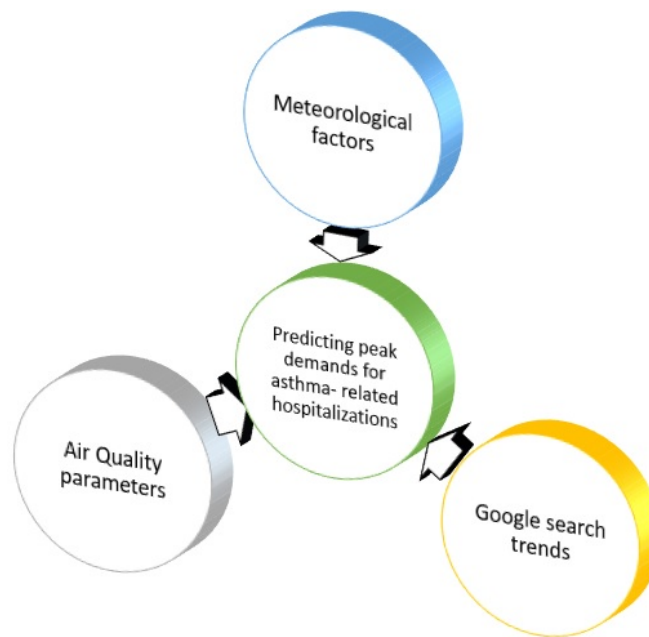


Figure 5.3: Risk assessment for asthma-related hospitalizations using diverse sources.

1. Data Collection
2. Data Preparation
3. Feature Selection
4. Prediction Model

The diagrammatic presentation of all the stages has been presented in Figure 5.4 and the detailed description of each stage has been covered underneath.

5.3.1 Data collection from diverse sources

The first step towards prediction modeling commences with collecting raw data. In this case, apart from asthma-related hospitalizations, the model involves data from distinct sources, air quality parameters, meteorological factors, and Google Trends. The list of attributes was enhanced with more attributes included that have not been part of any existing study, such as the Ultraviolet Index (UVI), soil deficit, dry bulb

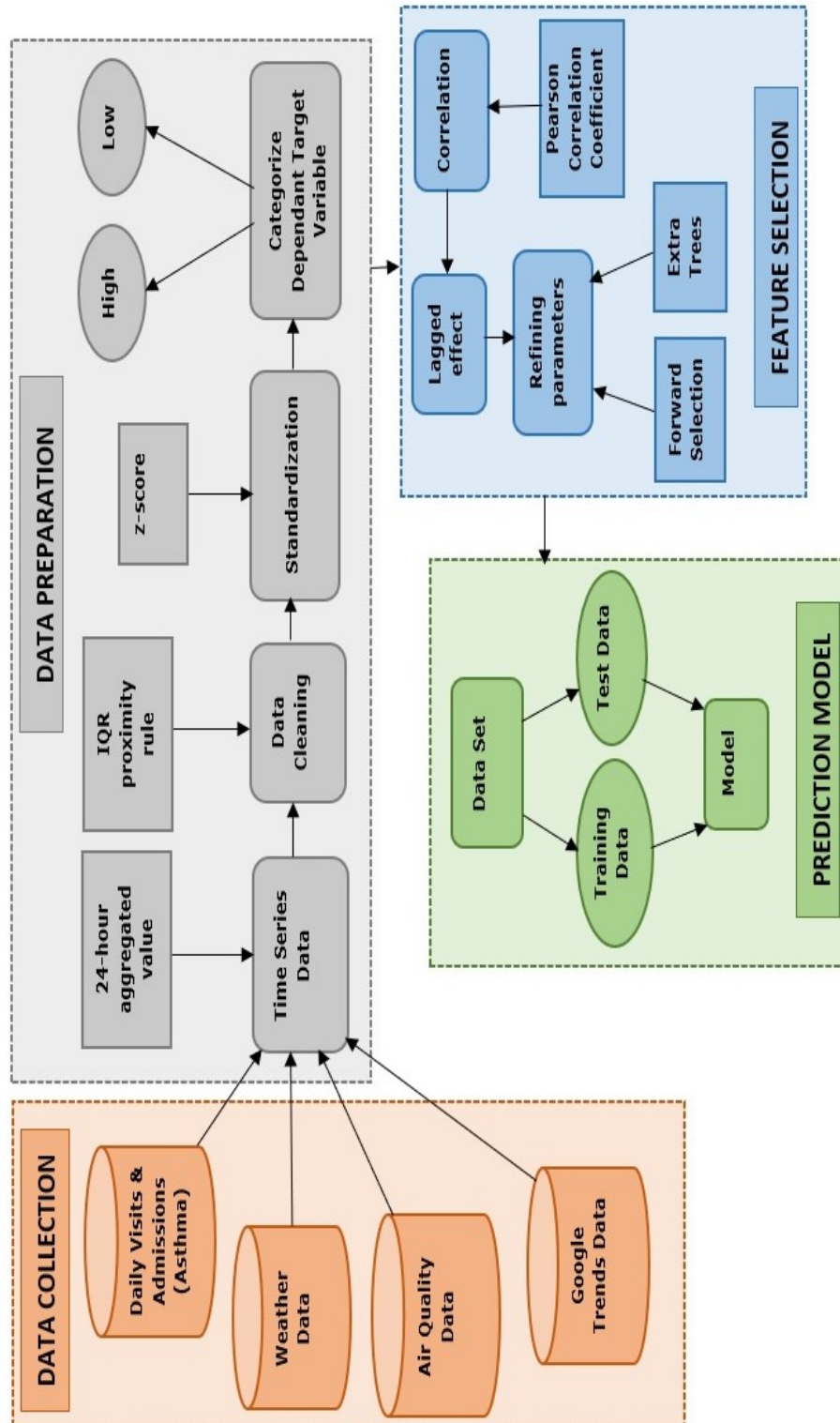


Figure 5.4: Processing heterogeneous data sources for generating the forecasting model.

temperature, wet bulb temperature, dew point temperature, and black carbon. Further, the collected data were preprocessed, prepared, and refined. Finally, it was used for predictive modeling for asthma.

Hospital data

De-identified data for asthma-related admissions and ED visits were collected from Auckland Hospital, Auckland District Health Board (ADHB), which is now known as Te Toka Tumai under the new health entity Te Whatu Ora. *ADHB* provides health services to over 494,000 residents living in the Auckland district and is the fourth largest and one of the fastest-growing DHBs in the country. Ethics approval for this study was obtained from *Auckland Health Research Ethics Committee*, AH22149, with an expiry date of 11/05/2024. The ethics approval letter has been added in Appendix B.1 for reference. The data for asthma was based on the International Classification of Disease Tenth edition -Australian Modification [ICD10-AM] codes *J45.2x to J45.90x*, *R05*, *R06.00-R06.2*. These classifications are used in the literature to identify asthma-related presentations to the hospital. This data accounts for daily counts of ED visits and admissions related to asthma for three years, starting from 1st Jan 2018 to 1st Jan 2021, forming a continuous data of 1097 days.

Meteorological data

The data for the meteorological factors were collected from the National Institute of Water and Atmospheric Research, commonly known as NIWA. The list of parameters that were inspected for this study were: temperature, wind, soil deficit², UVI, pressure, dry bulb temperature, wet bulb temperature, dew point temperature, and relative humidity. Temperature represents the aggregated daily readings and is calculated by taking the

²Soil moisture deficit, where full is 0 mm, and empty is 150 mm (AWC Available Water Capacity), refer National Climate Database: <https://cliflo.niwa.co.nz/>, for more details.

mean of the highest and lowest temperatures recorded in the specified period. On the other hand, the other temperature parameters, that is dry bulb, wet bulb, and dew point, were utilized to evaluate the humid air condition. They specify the actual temperature at a specific time when measurements are taken. Dry bulb temperature indicates heat content and refers to a temperature not affected by the air's moisture. In contrast, a moistened thermometer bulb exposed to airflow specifies a wet bulb thermometer. The third category of a temperature-related parameter is the dew point which defines the temperature at which air becomes completely saturated, more precisely the point at which water vapors precipitate out of the air. The soil moisture deficit provides updates on daily records in the rain table and gives information on water balance. It is calculated using incoming daily rainfall and indirectly symbolizes precipitation. NIWA maintains several sites for measuring all these parameters all over Auckland except UVI. For recording UVI, there are only five sites all over NZ. For our research, the site closest to Auckland, the one at *Leigh Marine Observatory*, was chosen for acquiring data for this parameter. The geographical distribution of sites used to record reading for these parameters has been represented in Figure 5.5. It can be seen that the site at Leigh is quite far from the Auckland region but is the nearest compared to all the other sites measuring UVI.

Air quality data

The air quality monitoring program comprises two different types of parameters:

1. Particulate matter which includes $PM_{2.5}$, PM_{10}
2. Gaseous Pollutants consisting of *oxides of nitrogen, carbon monoxide, sulfur dioxide, ozone*

Air quality data was collected from eight different sites dispersed over the region of Auckland, and these sites are managed by Auckland Council. The parameters that

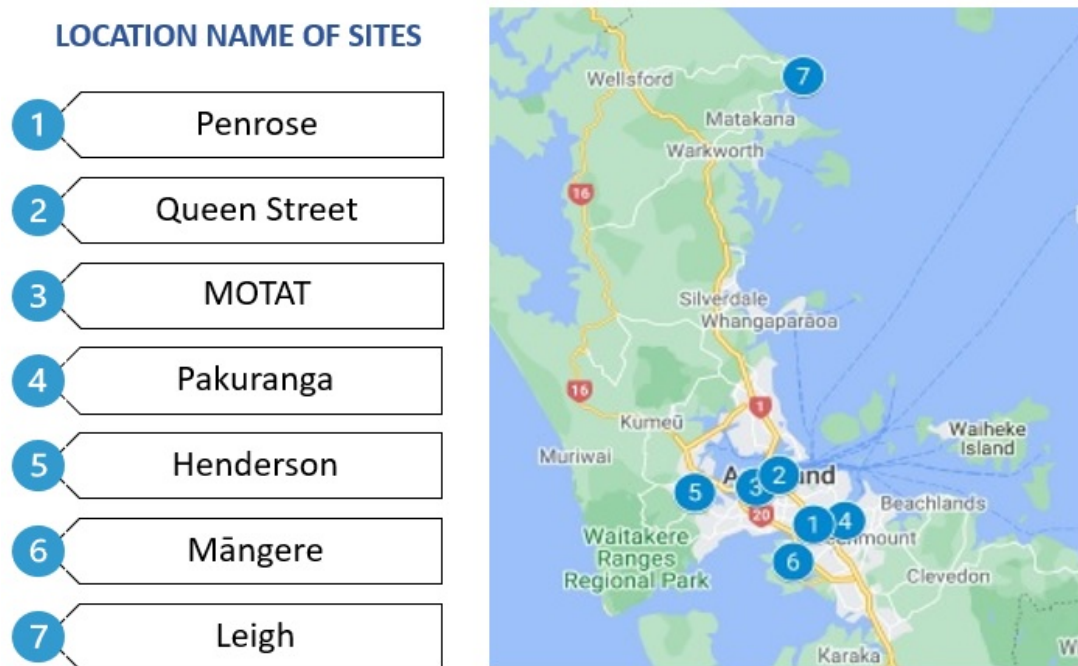


Figure 5.5: Geographic sites for gathering meteorological data.

were included under this category were: PM_{10} , $PM_{2.5}$, NO_x , NO_2 , *Ozone*, *Black Carbon* (880), *Black Carbon* (370), and SO_2 . The number with black carbon corresponds to the wavelength at which black carbon is measured by the device called an aethalometer. Several parameters were reported at only one site, like readings for black carbon parameter were available at only one site, which was at Henderson (represented as Site 2 in Figure 5.6). So this study does not signify the stratification related to spatial relevance. Figure 5.6 shows the distributed sites used for collecting air quality data dispersed all over the Auckland region.

Google Trends data

Google data can be used to tell stories. There has been work done in the past (Ram et al., 2015) that demonstrates the linkage between *social media data and disease case data*. Google is the generic choice and the most visited website worldwide, so we explored *Google Trends* as a social media resource. This dataset allowed examination of the

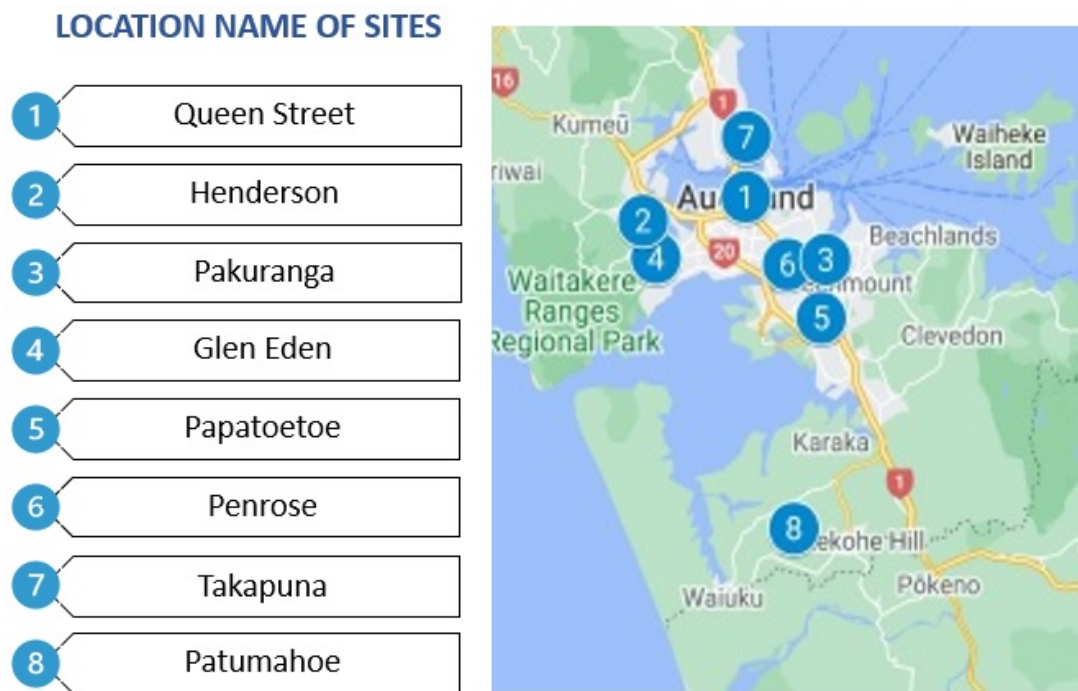


Figure 5.6: Representation of sites on the map used for collecting air quality data.

search trend and data under this category was collected on a daily basis and refined by keywords related to asthma in the Auckland region. The keywords used for refinement were: *respiratory, lungs, asthma, cough, chronic, oxygen, bacteria, inhaler, wheezing, puffer, and asthma attack*. These trends data, provide facts about how many searches related to asthma were performed each day in the Auckland region. Figure 5.7 shows the graph demonstrating daily search trends for Google, refined using asthma-allied keywords. It can be noticed that higher values are observed around July each year. June to August are the winter months in NZ and cold temperatures are believed to be a major trigger for asthma (Asthma & of America, 2017). Therefore, it can be inferred that a relationship might exist between search trends related to asthma and asthma-related hospitalizations.

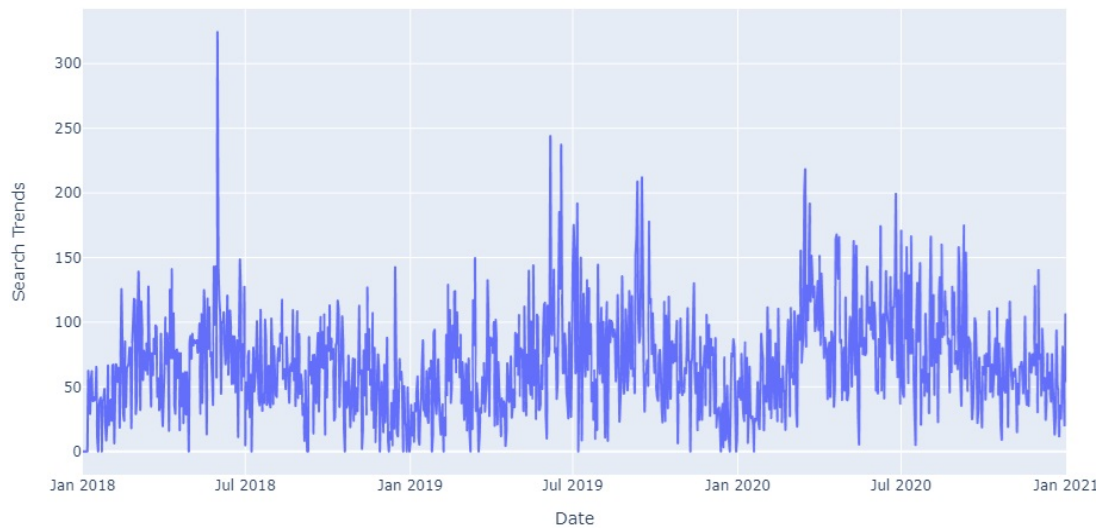


Figure 5.7: Graphs depicting the data values for Google search trends filtered using keywords related to asthma from a period of 1st Jan 2018 to 1st Jan 2021.

5.3.2 Data preparation

The statistics forming the hospital data for asthma-related visits and admissions are in the form of aggregated daily values. The missing entries were filtered based on dates absent from the hospital admissions and ED visits data. They were subsequently allocated a zero value, assuming there were no asthma-related admissions or hospitalizations on those days. The data for meteorological factors were also aggregated 24-hour values except for UVI, which were in the form of 10-minute data readings. In contrast, most of the parameters under the air quality category were hourly values. The readings collected with frequency in minutes or hours were agglomerated to obtain the 24-hour aggregated values accounting for continuous daily data. The Google Trends data were the daily search volume scaled and were comparable through time. The data summarized in Table 5.1 shows the frequency of measuring each attribute.

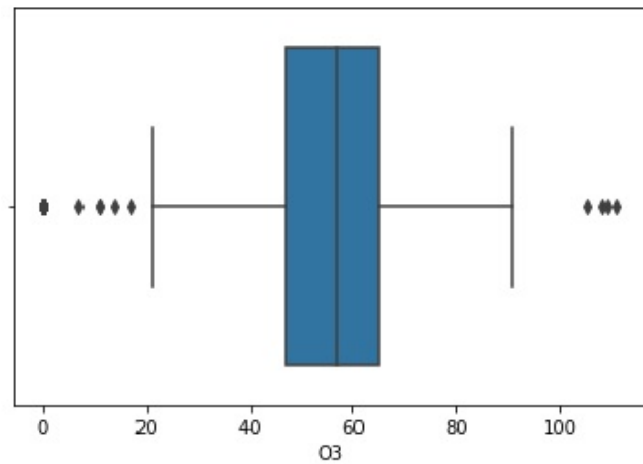
The aggregated value for every parameter, together with the date column, forms the respective dataset for every parameter, which is then cleaned from outliers. The outliers were detected using the *Inter-Quartile Range (IQR) proximity rule*, according to which the data points that fall below $Q1 - 1.5 \text{ IQR}$ or above $Q3 + 1.5 \text{ IQR}$ are outliers,

Table 5.1: Frequency of readings.

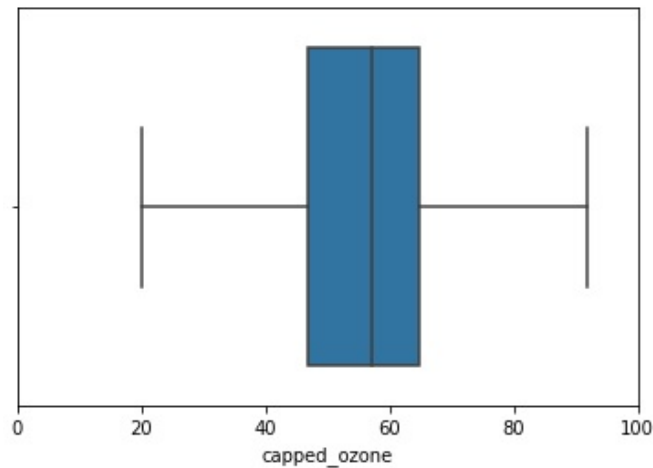
Parameter	Periodicity of Readings
UVI	10-Minute
Black Carbon (370)	Hourly
Black Carbon (880)	Hourly
NO _x	Hourly
NO ₂	Hourly
Ozone	Hourly
PM ₁₀	Hourly
PM _{2.5}	Hourly
SO ₂	Hourly
Dew Point Temperature	Daily
Dry Bulb Temperature	Daily
Hospital Data	Daily
Pressure	Daily
Relative Humidity	Daily
Soil Deficit	Daily
Temperature	Daily
Trends Data	Daily
Wet Bulb Temperature	Daily
Wind	Daily

where $Q1$, $Q3$ are the 25th and 75th percentile of the parameter value data column, and IQR represents the inter-quartile range and given by $Q3 - Q1$. The outliers detected are treated by capping them within the 25th and 75th percentile range. Figure 5.8a illustrates the state of a parameter named ozone with outliers, where big black dots represent outliers, and Figure 5.8b presents the plot after capping the outliers. The datasets corresponding to all the meteorological and air quality factors, plus Google Trends, form the predictors for predicting the target variable: *asthma-related ED visits and hospitalizations*.

The next step involves handling missing values for the input variables; *missing values* are defined as the data values that are not present for some variables in the dataset. There can be different ways to fill these values, such as i) deleting the records with missing values, ii) filling them with constant values, iii) using statistical methods



(a) Plotting of data values for ozone with outliers



(b) Plotting of data values for ozone after treating outliers

Figure 5.8: (a) Plot for Ozone with outliers (b) Plot for ozone after treatment of outliers.

like mean, median, mode, or linear regression iv) interpolation. The methods chosen depend on why those missing values may have arisen. As already stated, missing values for hospital data might have been due to the absence of admissions on a day, whereas the missing values corresponding to input predictor variables may be because of device failure, human errors, and others and are therefore filled using the mean function.

The data values for all the parameters were acquired from different sources and spanned varying degrees of magnitude, range, and units. Therefore, each parameter was scaled to acquire a standard range. The method we have used in this research for scaling

is *standardization*. A variable is scaled by subtracting the mean (called centering) and dividing by the standard deviation to shift the distribution to have a mean of zero and a standard deviation of 1. The standard score of sample x is calculated as:

$$z = \frac{x - \mu}{s}$$

where μ is the mean and s is the standard deviation.

Comparability has been drawn for one parameter in Figures. 5.9a and 5.9b concerning distribution for raw data and prepared data. To convert this problem into classification, we classified ED and hospitalization data related to asthma based on the standardized values into two categories; *High and Low*. *High* represents the days on which the number of patients was greater than the 70th percentile while the rest of the days were labeled as *Low*. Thus for this study, the 70th percentile was used as the threshold value.

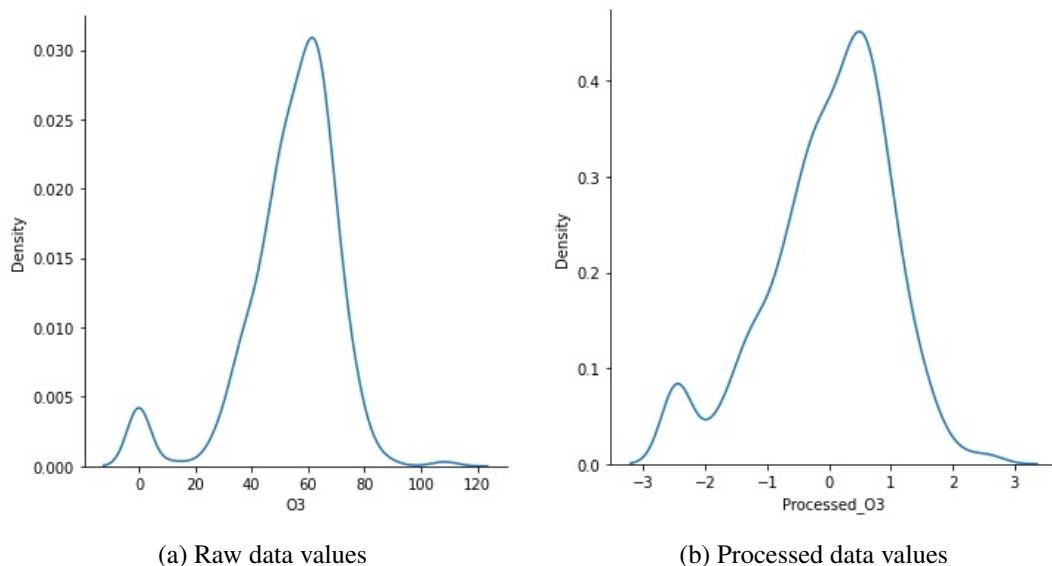


Figure 5.9: (a) Distribution of originally collected data for Ozone (b) Representation of distribution when (a) has been preprocessed and prepared.

5.3.3 Feature selection

In the first instance, an initial assessment was done to analyze the association between asthma-related ED visits and admissions data on one side and the predictor data on another side by using *Pearson's Correlation Coefficient or PCC*. It is known to be the best method for evaluating the strength of the statistical relationship between two continuous variables. The correlation value may range from -1 to +1, where -1 displays the *perfect negative relationship*, +1 shows a *perfect positive relationship*, and 0 shows *no relationship*. This correlation calculation does not depend on the units in which the variables are measured. To determine whether the correlation between the two variables is significant, the *p-value* obtained while calculating the correlation coefficient is compared to the significant level. Typically, a significant level expressed by α (alpha) equal to 0.05 is used. A *p-value* $\leq \alpha$ indicates that the correlation is statistically significant, while a *p-value* $> \alpha$ concludes that the correlation is not statistically significant.

The predictors witnessing significant correlation (according to *p-value*) with asthma-related ED visits and admissions were used as inputs for further processing. Changes in weather conditions or pollutants levels may not have an immediate impact on asthma conditions; instead, they may exacerbate asthma conditions leading to a delay between an environmental change and presentation to the ED or hospital. The values of these predictors on a particular day may not be linked with asthma-related ED visits and hospital admissions on the same day due to this delay. The lagged effect was applied to predictor variables to achieve the best representation between predictors and the target variable. The lagged effect of UVI has been displayed in Figure 5.10 for 0 to 4 days, where 0 represents the same-day effect. It can be seen that UVI has the maximum effect on the third day. The lagged period of all the parameters has been illustrated in Figure 5.11. A blank space following a parameter signifies a lag of zero days.

The next step towards developing a predictive model was selecting the most crucial and relevant features from the vast list of features in a given dataset, thereby reducing the complexity of a model and making it easier to interpret. Basically, two methods were employed for feature selection: *forward feature selection* and *Extra Trees*. In the forward feature selection method, we start with the null model having no covariates. At each algorithm step, a new variable is added to the current model based on criteria like improving accuracy. On the other hand, bagged decision trees, that is, Extra Trees, work by estimating the importance of features.

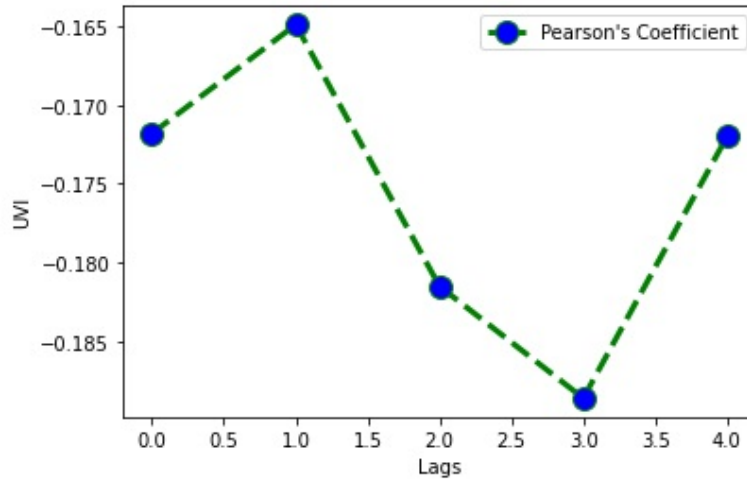


Figure 5.10: Lagged effect of UVI for a period of 0 to 4 days.

5.3.4 Prediction model

Finally, the date and number of patients columns were dropped to obtain a pair representing the complete dataset in the form (X, y) . Here, X represents the predictor variables and is a matrix of size (1097×14) , while y is the class variable representing the number of asthma-related visits and hospitalizations either as Low or High. The ratio of samples for the Low to High class was 866 to 231. The numeric value used for class *Low* was 0 while for *High* was 1. The complete dataset was divided into training and test sets in a ratio of 50:50.

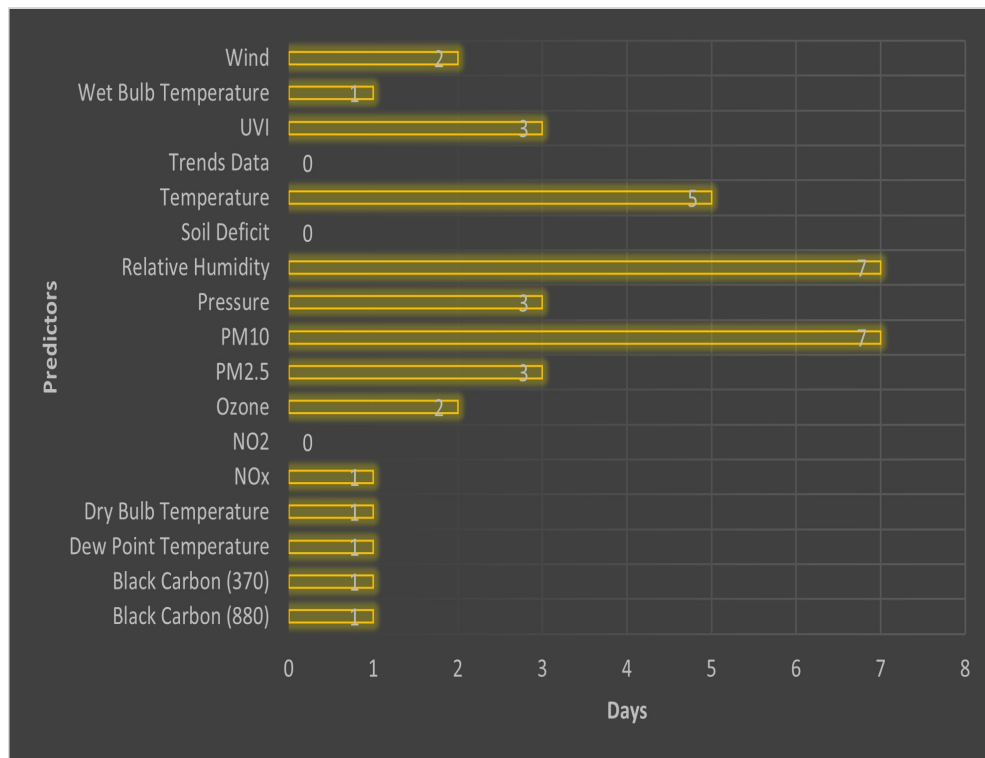


Figure 5.11: Days as a unit of lags.

The prediction modeling process was further divided into sub-phases according to three different methodologies used for prediction modeling, which are discussed below:

Method 1: Using standard machine learning algorithms.

In the first phase, four standard machine learning algorithms were executed to obtain the prediction model for asthma. These techniques were *MLP*, *RF*, *SVM*, and *XGB*. *MLP* is a feed-forward ANN model that is used for mapping a set of input variables to the appropriate output. An MLP model comprises several layers that are fully connected; essentially, there is an input layer, an output layer, and between them, there may be one or more hidden layers. For this study, a three-layered MLP algorithm trained using back-propagation was modeled with an input layer, a hidden layer, and an output layer. MLP classifiers rely on an underlying neural network to perform the classification task. An *RF* algorithm consists of many decision trees and produces the outcome based on the prediction

of the decision trees. The final output is generated by taking the average or mean of the output from various trees. *SVM* is a fast and dependable classification algorithm that performs very well with a limited amount of data to analyze. The goal of the *SVM* algorithm is to create the best line or decision boundary that can split up *n*-dimensional space into classes so that the new data points can be classified in the correct category. *XGB* is a boosting algorithm that is based on gradient-boosted decision trees. It induces a prediction model by generating an ensemble of weak prediction models, decision trees in this case. *XGB* is also known as a regularized boosting method that helps in reducing over-fitting. It has its roots in the decision tree and uses boosting on top of it to improve performance. It is believed to be comparatively faster and more efficient than older ensemble classifiers.

Method 2: Using an ensemble by combining different algorithms.

In the second phase, an ensemble classifier was developed by combining three classifiers as building blocks, as illustrated in Figure 5.12. Thus, the three classifiers' predictions act as inputs into the ensemble, which produces the final prediction by employing the *majority voting* mechanism. Out of three, two classifiers were built using *XGB* classifier algorithms with distinctive parameter configurations. The reason for choosing *XGB* in this ensemble modeling will be explained in the next section. The third model whose predictions were used in this ensemble was obtained by applying Autoregressive Integrated Moving Average (*ARIMA*) algorithm. In order to check the usability of past hospitalizations data values for predicting future demands, the *ARIMA* model was chosen as the third input for the ensemble model. This algorithm is based on statistical analysis that utilizes time series data values to understand the data patterns, and by using the past values, it produces future predictions. Here, the *ARIMA* classifier was

trained using the past values of asthma-related hospitalizations. The input for the ARIMA model was the actual number of daily asthma-related hospitalizations instead of the target class. As a result, this model predicts future values for the number of asthma-related hospitalizations, which were then converted into suitable classes using the 70th percentile, as mentioned above.

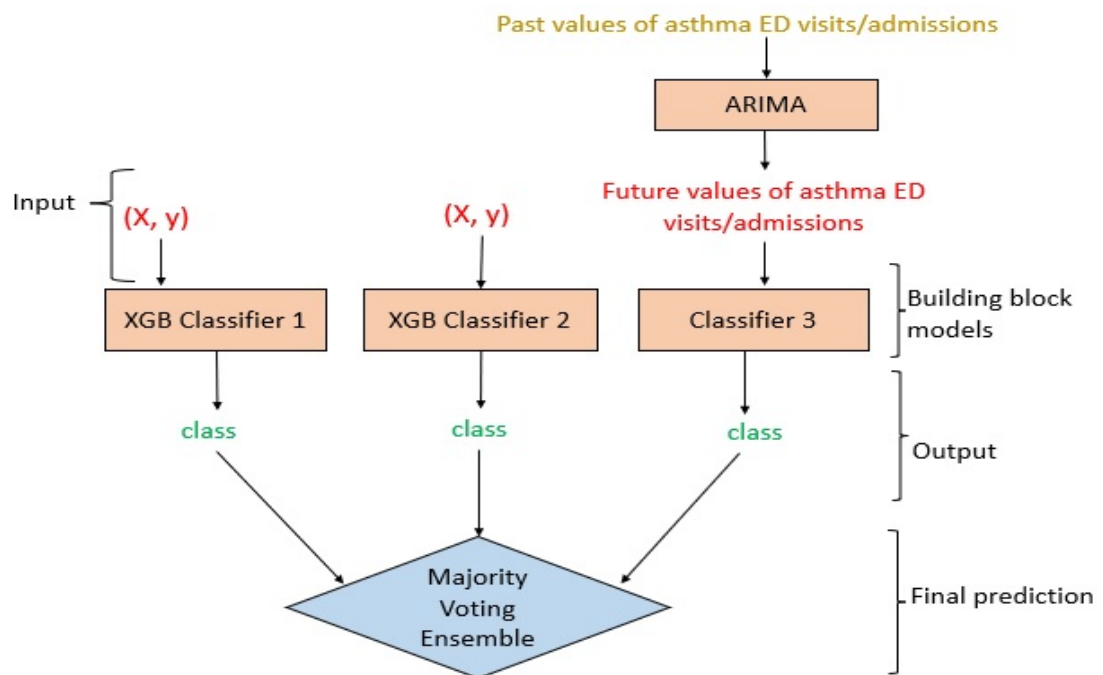


Figure 5.12: Ensemble modeling using three base models with final classification generated using majority voting mechanism.

Method 3: Using standard machine learning algorithms on over-sampled dataset.

The above-discussed methodologies *Method 1* and *Method 2*, were implemented on the dataset where the number of instances under classes 0 and 1 were 866 and 231, respectively. Therefore, to obtain a fair, balanced classification scenario, over-sampling techniques were applied to generate synthetic data. Subsequently, two methodologies, SMOTE and random over-sampling were utilized to obtain a balanced dataset concerning the above-specified classes, viz. 0 and 1. *SMOTE* is

a class of oversampling techniques that generate new minority instances by linearly interpolating between the existing minority instances. It randomly chooses k -nearest neighbors in the minority class for synthesizing new records. *Random over-sampling* is another method that over-samples the minority instances by adding more copies to the minority class. When there is a dearth of data, oversampling proves to be a good choice (Dattagupta, 2018). After obtaining a balanced dataset, an array of standard machine learning algorithms (MLP, RF, SVM, and XGB) listed in Method 1, were executed on this balanced dataset.

5.4 Experimental Evaluation

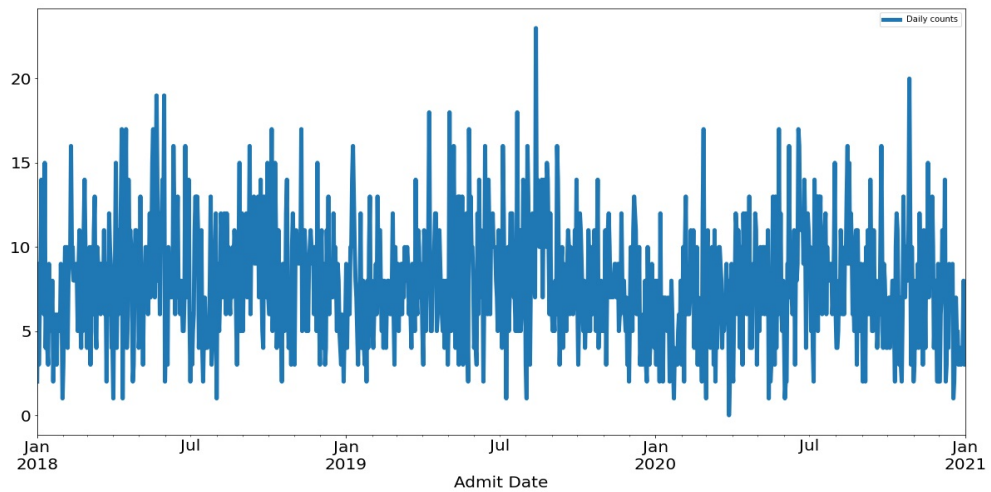
The experiments for all the methodologies discussed in this chapter can be accessed through the GitHub repository; check Appendix C for more details. All the experiments were performed on a 64-bit Windows running on a 4×3.40 GHz Intel Core i5 CPU with 16 GB of memory. The programming was done using machine learning packages available in Python. The spread of asthma-related daily counts of ED visits and admissions for the study period from 1st Jan 2018 to 1st Jan 2021 has been represented in Figure 5.13a. The visualizations to analyze the hospital rate was simplified by converting the daily visits and admissions data into monthly counts using rolling mean with a window size of seven days, Figure 5.13b shows the monthly rate of asthma-related ED visits and admissions by taking the rolling mean with a window size of 7 days. Next, the ARIMA model was applied to predict the monthly visits and admissions based on the past values of the same attribute. For executing this model, half of the data was used for training, and the other half was used for testing. Hence, the model was trained from 01/2018 to 06/2019 and was tested from 07/2019 to 12/2020. Figure 5.13c shows the actual values for the monthly hospital data with the predicted values obtained from ARIMA for the test period. We can witness a significant gap between actual

and predicted values, stating that using just the past values of asthma-related ED and admissions trends is inadequate to predict future values. Therefore, we inspected the utilization of heterogeneous data sources for predicting the daily counts of ED visits and admissions related to asthma.

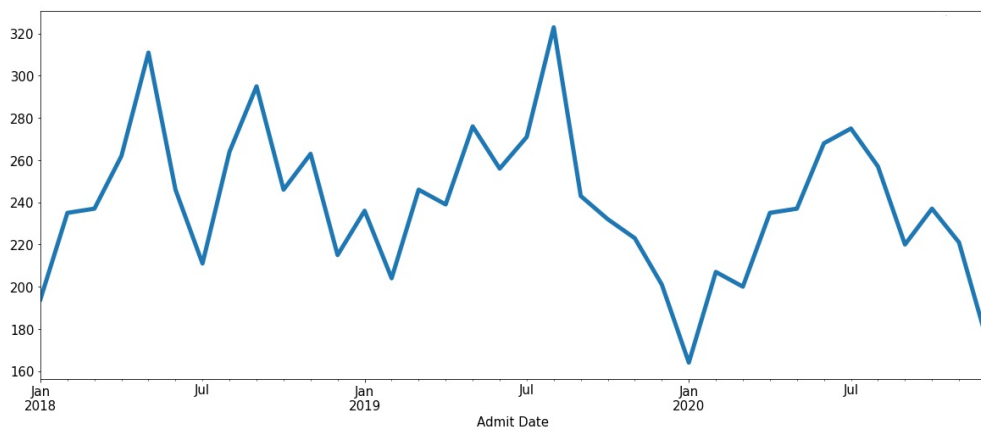
The results obtained after calculating the PCC have been analyzed in Figure 5.14. These results have been produced after applying the lagged effect. All the coefficient values were significant at the 0.01 level except SO₂, which managed to achieve a correlation value of 0.0628 but with a p-value of 0.0377. As the p-value for this correlation was insignificant, so SO₂ was removed from the list of predictors for further processing. The results in Table 5.2 depict the analytical summary of all the selected parameters. The second and third columns express the *mean* and *standard deviation* of the data values for each parameter. From the results in this table, we could witness a small correlation value of asthma-related visits and admissions with PM₁₀, Pressure, and NO_x. Even though the correlation coefficient value was small for these attributes, dropping them from the prediction model produced unfavorable consequences on accuracy. Therefore, these variables were included to develop the prediction model. Based upon the results of feature selection methods, three attributes were dropped from the predictor list, which were: NO₂, Relative Humidity, and Dry Bulb Temperature, as these parameters seemed irrelevant for modeling.

The execution of the prediction model listed under the three methodologies in the previous section has been assessed in terms of three different metrics: *Accuracy*, *Precision*, and *Recall*. All of them are reported using True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). *Accuracy* is the ratio of the total number of correct predictions to the total number of predictions.

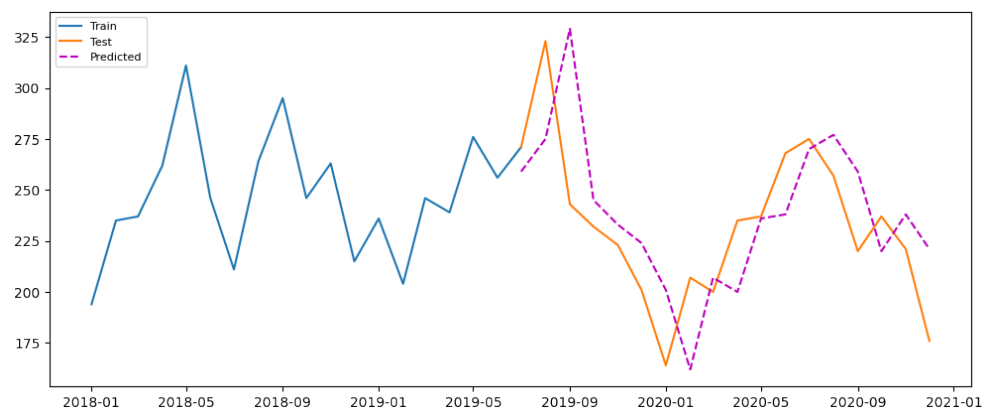
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$



(a) Plot for daily hospital data values



(b) Monthly data values obtained by applying rolling mean function to daily data values in a)



(c) Visual representation for actual v/s predicted graphs

Figure 5.13: a) Plot for daily counts of asthma-related ED visits and admissions b) Monthly visualizations of the hospital data from 01/2018 to 12/2020 c) Timeline for actual and predicted values for monthly rate of ED visits and admissions using ARIMA for a test period from 07/2019 to 12/2020.

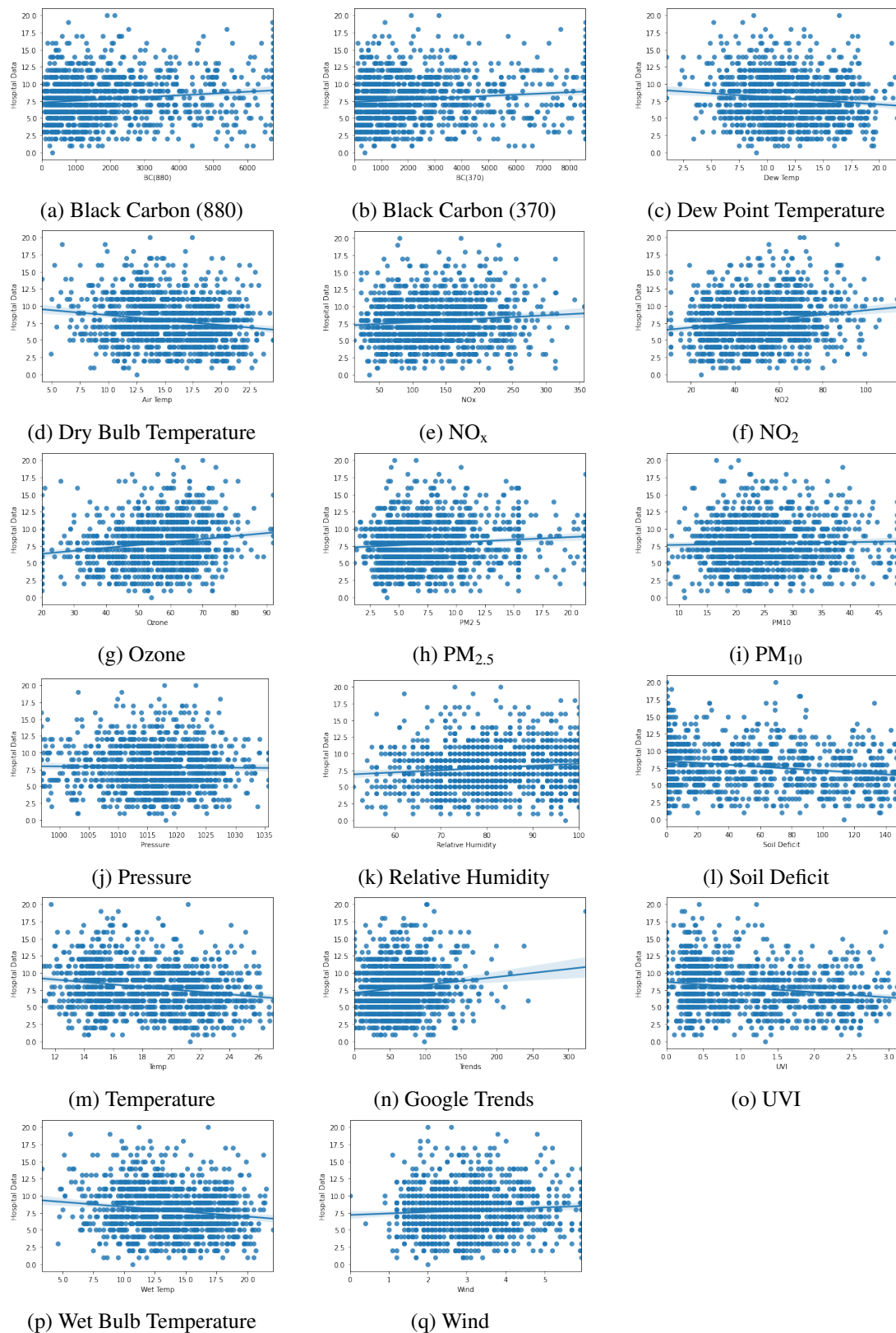


Figure 5.14: Diagrammatic representation of the correlation between predictors and daily counts of asthma-related hospitalizations.

Table 5.2: Analytical summary of predictors.

Parameter	Mean	Std Dev	Correlation
Black Carbon (880)	2194.78	1873.76	0.1433
Black Carbon (370)	2788.81	2581.86	0.1348
Dew Point Temperature	12.09	3.75	-0.1184
Dry Bulb Temperature	15.51	3.88	-0.1642
NO _x	135.00	64.69	0.0962
NO ₂	51.77	18.97	0.1734
Ozone	54.89	14.15	0.1788
PM _{2.5}	8.14	4.31	0.1008
PM ₁₀	25.69	8.19	0.0326
Pressure	1016.05	7.69	-0.0154
Relative Humidity	80.90	11.44	0.1108
Soil Deficit	47.26	47.77	-0.1963
Temperature	18.45	3.65	-0.1923
Trends Data	67.82	39.39	0.1314
UVI	1.09	0.85	-0.1887
Wet Bulb Temperature	13.65	3.49	-0.1451
Wind	3.06	1.17	0.0731

Precision is the ability of the classifier not to label a negative sample as positive. In simple terms, it is the ratio between true positives and all the positives.

$$Precision = \frac{TP}{TP + FP}$$

Recall accounts for the competency of the classifier to find all the positive samples. It measures how accurately the model can identify the relevant data. It is also referred to as Sensitivity or True Positive Rate. Mathematically,

$$Recall = \frac{TP}{TP + FN}$$

The performance of all the standard machine learning algorithms covered in *Method 1* (viz MLP, RF, SVM, and XGB) has been demonstrated in Table 5.3. It can be seen that out of four basic models, *XGB Classifiers* perform the best for all three measures.

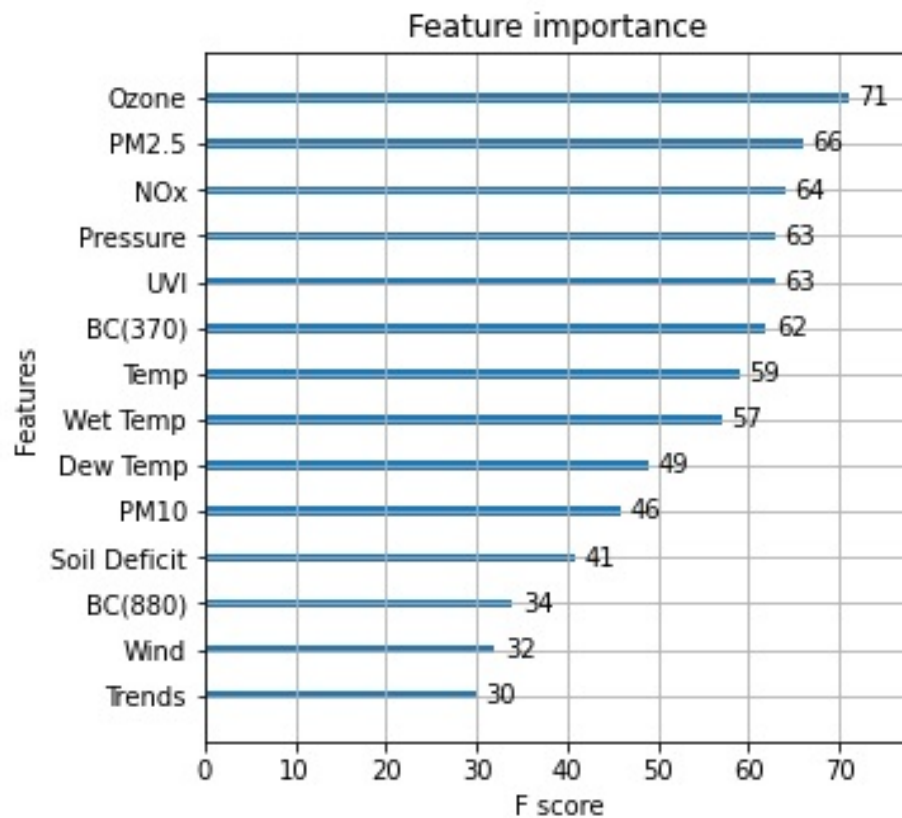


Figure 5.15: Feature importance by XGB classifiers.

All the algorithms have comparable accuracy, but XGB classifiers outperformed with regard to precision. XGB classifiers require the numerical features to be scaled before using them in the model. One way to visualize the XGBoost model is by depicting the importance of each feature, as shown in Figure 5.15.

Besides this, two diagnostic tools that assist in evaluating the predictive models

Table 5.3: Comparative analysis of different Machine Learning algorithms.

Model	Accuracy (%)	Precision (%)	Recall (%)
MLP	78.51	76.02	78.51
RF	78.14	72.32	78.14
SVM	78.14	61.06	78.14
XGB	78.87	79.81	78.87
Ensemble	79.05	80.44	79.05

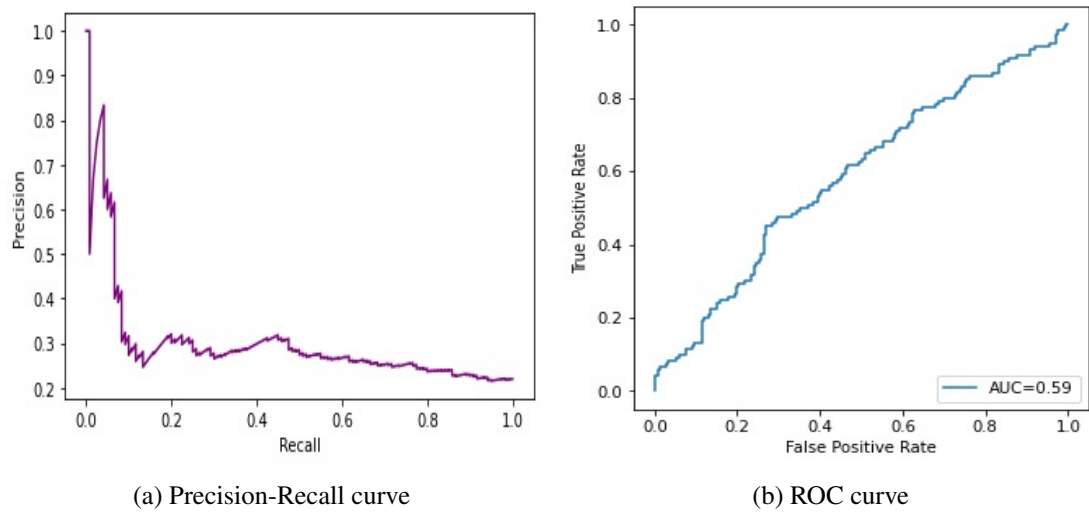


Figure 5.16: a) Precision-Recall curve b) ROC curve for XGB classifier.

for binary classification problems are the *Precision-Recall curve* and *ROC curve*. The trade-off between precision and recall is portrayed using the *Precision-Recall curve* for the different threshold values. A higher area under the curve depicts a high value for recall and precision. *ROC*, also known as the Receiver Operating Characteristics curve, presents the trade-off between true positive and false positive rates for different probability threshold values. Precision-Recall curves are more useful when classes are imbalanced, whereas ROC curves are more appropriate for balanced classes. Figure 5.16 displays the Precision-Recall and ROC curve for XBG classifier. Another metric used to assess a model's accuracy was *AUC*, which specifies the area under the curve. An *excellent model* tends to obtain an AUC close to 1, while a 0 value of AUC signifies a *poor model*.

The results obtained on implementing ensemble modeling discussed as *Method 2* for predictive modeling in the previous section have also been presented in Table 5.3. This ensemble model was built using two XGB classifiers plus a classifier using output from the ARIMA model. The reason for choosing XGB classifiers was their best performance out of all the algorithms executed in Method 1. The majority ensemble witnessed better performance than all the models generated under Method 1. This improvement is due

Table 5.4: Comparison of performance achieved through implementing different algorithms over-sampled data.

Model	SMOTE			Random Over-Sampling		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
MLP	70.79	70.85	70.79	80.37	81.27	80.37
RF	78.87	78.87	78.87	86.37	86.45	86.37
SVM	75.87	83.72	75.87	91.00	92.37	91.00
XGB	73.44	73.48	73.44	81.76	82.17	81.76

to the competency of ensemble modeling to decrease the number of misclassifications and boost the accuracy of the classifier.

The *Method 3* discussed in the previous section was dedicated to testing the over-sampling techniques. This section portrays the results obtained by over-sampling. As mentioned above, SMOTE and random over-sampling were used to generate data to acquire a balanced dataset. With both of these techniques, the minority class, *High*, was over-sampled to 866 instances, thus achieving equal number of records under both classes. Figure 5.17 shows the distribution of data points under both the classes for UVI parameter, where Figure 5.17a represents the plot for the initial form of data with *no oversampling*, and Figures 5.17b, 5.17c shows the plots for *over-sampled data with SMOTE and random over-sampling respectively*.

The diverse behavior in producing over-sampled data can be seen under both methods (in Figures 5.17b and 5.17c); thus, the classification results produced by the respective dataset generated by these two methods also differ. Table 5.4 presents the performance comparison between the two different balanced datasets generated by SMOTE and random over-sampling. The results show that the model built on random over-sampled data performs better than the data generated through SMOTE. Likewise, it can be noticed that SVM implemented on the random over-sampled data produces the best results out of all the methods discussed so far. The reason why SVM couldn't perform well on the original dataset, where the ratio of instances under classes 0 and 1

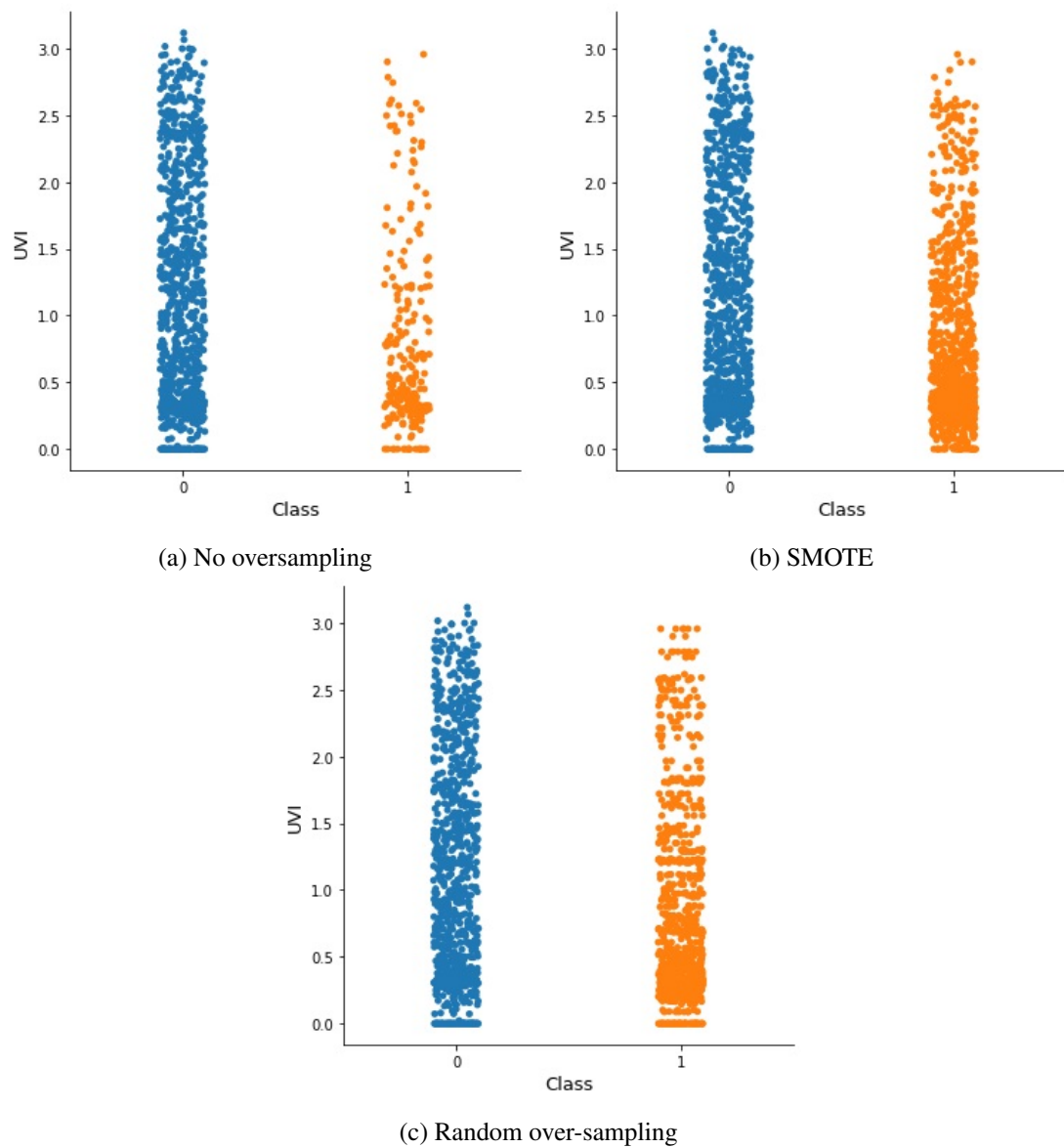


Figure 5.17: Distribution of data points with a) no oversampling, b) SMOTE, c) random over-sampling.

was 866 and 231, was due to their tendency to predict the majority class as the number of minority instances was meager in that dataset. Neural networks and boosting algorithms can overcome the situation when there is a big difference in the number of instances under the classes. Figure 5.18 shows the Precision-Recall and ROC curve for a balanced dataset using SVM. It can be noticed that the graphs here manifest exemplary model conduct than the performance depicted in Figure 5.16, achieved through an unbalanced

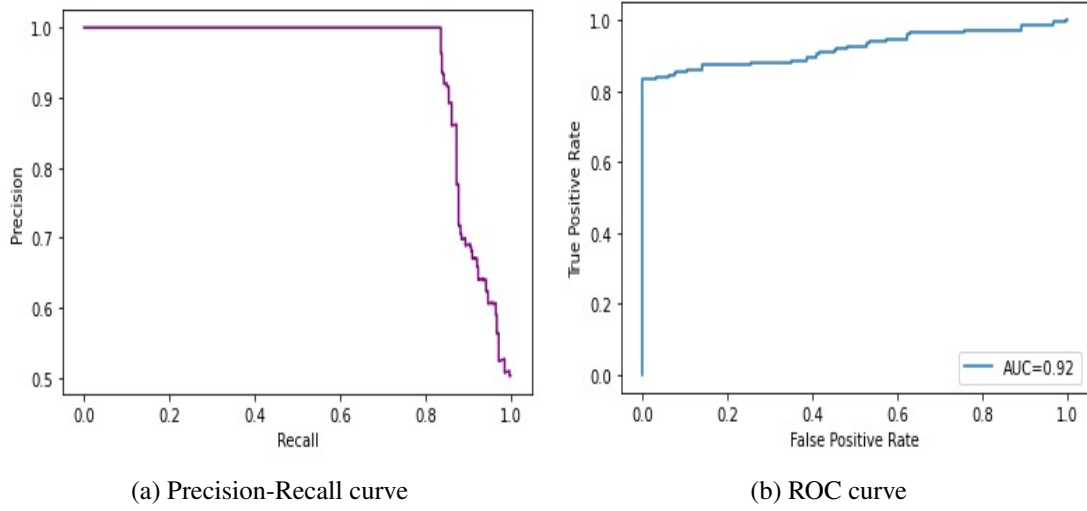


Figure 5.18: a) Precision-Recall curve b) ROC curve for a balanced dataset using SVM.

dataset.

Additional work was done to check over-fitting in the case of the SVM model generated on random over-sampled data. The learning curves were plotted for training and cross-validation scores for an SVM model implemented on a random over-sampled dataset. The learning plots in Figure 5.19 show that with increased training set size, the validation scores improve. The gap between training and validation scores initializes to a high value, which witnesses a declining trend with expanding training set size. This gap reduces to the least value at the end, experiencing a nice fit. Hence, in a balanced dataset, an SVM model shows the competency to provide the best classification accuracy among all other methods discussed in this chapter.

A point that can be highlighted here, is that recall and accuracy share the same value for all cases. This is likely due to the use of *weighted recall*. If we have two classes, a weighted recall would be calculated as below:

$$Weighted_r = \frac{(r_{c1} * |c1|) + (r_{c2} * |c2|)}{|c1| + |c2|}$$

where r_{c1} and r_{c2} are the recall for *class 1* and *class 2*, and $|c1|$ and $|c2|$ are the number

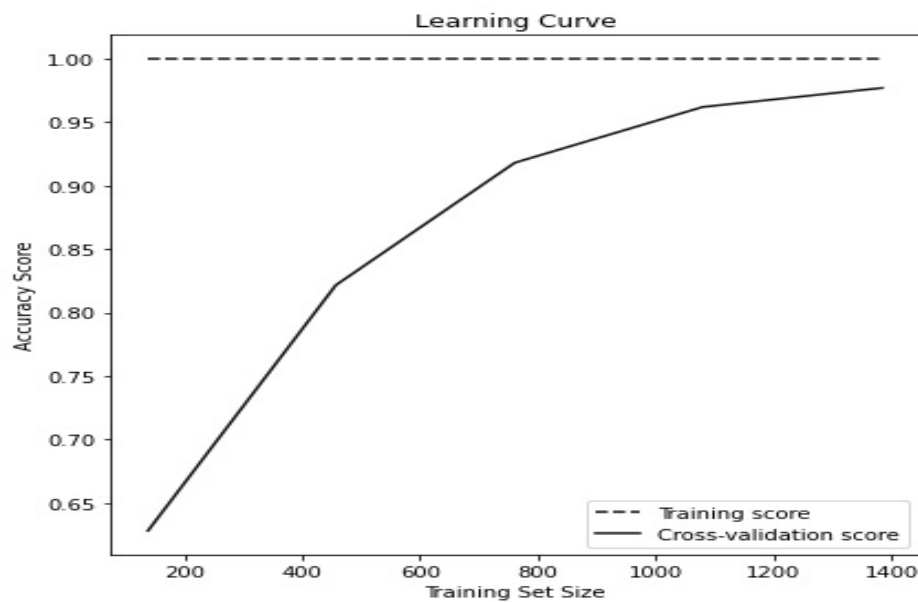


Figure 5.19: Learning curves for training and cross-validation phases.

of records in class 1 and class 2. In the above-defined formulas, a positive recall would be $TP/(TP + FN)$, whereas $TN/(TN + FP)$ will be a negative recall. $|c1|$ and $|c2|$ would hold the value $(TP + FN)$, $(TN + FP)$ respectively. Substituting these values to the formula of weighted recall would give a value equal to accuracy.

5.5 Applicability and Limitations

Our predictive modeling is novel as it uses multiple sources as input and real hospital data values from ADHB and this may serve as a key method to minimize the impact of asthma through accurate risk prediction of peak demand for asthma in hospitals. No other study has used combination of all these parameters to develop prediction modeling for asthma, specifically in NZ. It reveals the prominence of diverse data sources for the early prediction of days on which asthma-related ED visits and hospital admissions are greater than the threshold value. Generating a model leveraging diverse data sources to predict adverse events, like overcrowding related to asthma hospitalization at a

population level, would provide timely and targeted influences to minimize the socio-economic impact of asthma. Moreover, applicability of multi-models and multiple sampling techniques has been shown for generating prediction modeling. The high accuracy achieved through this prediction model can significantly help in planning for peak demands related to asthma and may prevent deaths, thus ensuring an enhanced healthcare delivery. Although we have split hospital data into two classes, Low and High, using the 70th percentile as the threshold value, this division can be adjusted according to required standards. Additionally, this prediction modeling can also focus on cutting down the monetary loss to the economy.

Even though the predictor data set has been enriched with many parameters, it still lacks a principal factor, that is, *pollen*. It is one of the most common triggers that can provoke asthma exacerbations. However, NZ does not have a national monitoring system to detect pollen counts, thus preventing the inclusion of this quantitative measure as a predictor in our study. Additionally, the ED visits and hospitalization data come only from one DHB within NZ and may not be generalizable to all NZ. Nevertheless, Auckland does have a similar deprivation profile to the rest of NZ and is the fourth largest DHB in NZ, thus providing a useful initial dataset for this work. Furthermore, individuals who do not have access to technological usage may not be represented in the Google Trends data.

5.6 Chapter Summary

The work discussed in this chapter is the first study of its kind to explore population health prediction modeling in asthma, which has been a prominent health issue in NZ. This study provides an essential foundation for future work. Under this study, we have explored different data sources and assessed the risk associated with asthma, producing an accurate and precise prediction tool, using a refined heterogeneous set

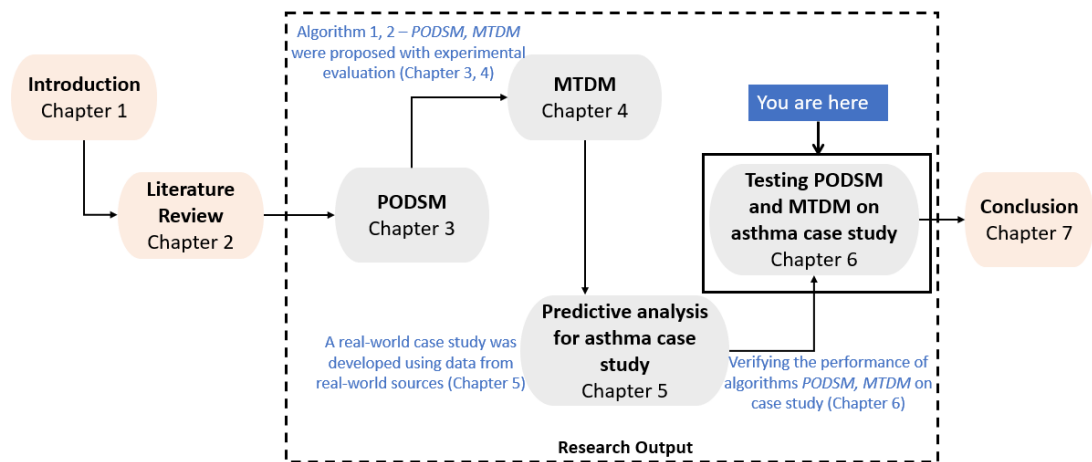
of parameters originating from distributed sites. Moreover, this work examines the effect of novel sources like UVI on asthma. A staged procedure commencing from data collection to prediction modeling has also been examined in this chapter. Also, we have demonstrated execution of multiple machine learning algorithms and variations of sampling techniques, therefore generating thorough analysis and experimentation of prediction modeling.

The major motive behind this work was to build a real dataset using heterogeneous data streams that could validate the performance of PODSM and MTDM. The ideology of applying differential modeling techniques was to generate predictive analytics and check the model's efficacy in an offline context. The maximum accuracy in predicting peak demand days for asthma-related visits and admissions in Auckland is 91.00%, with precision and recall holding values of 92.37% and 91.00%, respectively. These techniques can be utilized as a prominent planning strategy for public health providers in the event of excessive demands; such as overcrowding which is usually witnessed specifically in cold weather. Similar experiences were realized during COVID-19 outbreaks.

Chapter 6

Implementation of PODSM and MTDM on the asthma case study

The main goal of this chapter is to validate the competency of novel algorithms - PODSM and MTDM on the asthma case study that uses the real dataset. It will discuss the parameter configuration that is required for executing these algorithms. In addition to reporting the results obtained by executing these algorithms, this chapter also works toward testing the parameter sensitivity of the algorithms.



Roadmap of the thesis.

6.1 Introduction

We have seen in Section 5.4 modeling based on past hospital data does not necessarily serve as an optimal way to predict future outcomes. Through the work done in the case study covering the asthma domain, we aim to exploit diverse sources that can affect asthma conditions. The most crucial question that arises while executing these heterogeneous sources is *how to integrate multiple sources or data streams* (Sarkar et al., 2014). As an outcome, a model taking advantage of all these data streams would provide an online classification of peak demand days for asthma-related hospitalizations into *High or Low*. Also, this context would provide a perfect scenario to validate the working of PODSM and MTDM in a real case study.

This chapter is focused on implementing an online decentralized architecture for predicting asthma-related hospital demands in Auckland, NZ, in near-real time. Such predictive analytics could do online processing for the prediction of peak demand days for asthma which can help healthcare providers plan for resource allocation without impacting performance and safety. The decentralized distributed architecture allows the *conjoint processing of heterogeneous data sources* without data centralization to refine the prediction model for asthma. These sources account for triggers that exacerbate asthma conditions comprising data for meteorological and air quality factors and the data on Google Trends as discussed in Section 5.3.1. Exploiting such an architecture that encourages distributed processing of diverse sources rather than centralizing data at a single site has the potential to stave off the requirement of concentrating all the data at a single location. This chapter includes an experimentation comparison between centralized and decentralized architectures in a streaming context. Under decentralized architecture, state-of-the-art algorithms discussed in Section 2.6 were executed with a major emphasis on PODSM and MTDM.

6.2 Problem Definition

ED overcrowding is a global healthcare concern that requires short-term decision support in urgent and emergency healthcare. According to a recent report, with COVID-19 infection control, an increase in ED patients by three and four percent each year were predictable (Witton, 2019). This overcrowding problem requires adaption to changes in demand in real-time or near-real-time without disrupting its normal operation. NZ is a place witnessing inconsistent weather conditions, which causes uncertainty in the number of admissions and ED visits due to asthma. Extensive literature suggests a strong relationship between asthma exacerbation and temperature. Also, in Section 5.3.1, we have proved a good correlation between asthma-related ED visits and hospitalizations, and temperature readings. Additionally, NZ experiences uncertain variations in other parameters which affect asthma conditions, such as wind and precipitation along with the temperature. Through the correlation study presented in Chapter 5, we also found a suitable correlation between asthma hospital rates and UVI, which has hardly been reported in any existing study. At the population level, the current asthma disease surveillance programs rely on weekly reports based on data collected from various local resources by health departments. Such data have a lag time of weeks, providing retrospective information that is not amenable to proactive and timely preventive interventions. However, accurate and timely surveillance data could allow for prompt and targeted interventions at the community level.

As such, the existing prediction model for asthma (Ram et al., 2015), (Khatri & Tamil, 2017) using heterogeneous data streams requires centralization of complete data from all the sources in offline mode, thereby involving colossal costs in the form of communication and computation. The offline mode of learning requires a complete dataset to be statically available. The model thus generated from such training methods is not able to learn in real-time and usually requires retraining from scratch for new data.

In this chapter, we are working towards online processing of heterogeneous sources for developing predictive modeling for asthma. With the online mode of learning, the model or algorithms have the ability to adapt dynamically to new patterns in data. The models generated from online learning don't require an entire dataset for learning. Instead, they learn incrementally from new incoming data in real time as they arrive. Thus, they are more efficient and flexible for solving online classification tasks in a variety of real-world data mining applications. Due to uncertain changes in input predictors, like weather conditions used for predicting asthma-related ED visits and hospitalizations, the requirement is to implement efficient and scalable algorithms that have the ability to adapt to underlying changes in patterns seen in the data.

Moreover, instead of centralizing data from diverse sources, this chapter pinpoints the processing of dispersed heterogeneous sources that restrains the need to centralize all the data for developing a prediction model for asthma, thereby saving resources. Under the decentralized distributed implementation in the streaming context, a comparative analysis has been conducted to compare the performance of state-of-the-art methods comprising distributed ensembles, Parks' method, and HDSM as described in Section 2.6. On top of this architecture, work has been done for implementing two algorithms - PODSM and MTDM, discussed in Sections 3.3.1 and 4.3.1, respectively, which are targeted toward economizing the prediction process.

6.3 Proposed Methodology

The two architectures used for processing distributed heterogeneous sources are centralized and decentralized, as shown in Figure 1.1. In this chapter, we will demonstrate the usage of both of them for asthma predictive modeling in a streaming context. Besides this, work has also been done for executing and comparing different algorithms for generating an asthma prediction model using an online decentralized framework. The

different algorithms that would be tested on an asthma case study have been demonstrated in Figure 6.1. The inputs to all these methodologies would be originating from distributed heterogeneous sources and the application of these algorithms would provide a class value as High or Low, representing asthma-related ED visits and hospitalizations. An effort has been made to analyze the comparison between different algorithms and summarize their characteristics. As mentioned earlier, the geographic area under study is Auckland, so we would obtain a model for Auckland after analyzing the correlation of an array of parameters with hospital data for this region. The primary emphasis of this chapter would be testing distributed miner algorithms, specifically PODSM and MTDM, on the asthma dataset. This is to check what effect these algorithms might have on all the metrics and whether they can attain a trade-off between *accuracy and resources* on this dataset.

All the methods implemented in Section 5.3.4 for developing the prediction model for asthma highlight the centralized approach for processing heterogeneously distributed sites and that too in an offline context. There was a total of 14 different parameters that were used for prediction modeling under all the methods discussed in Section 5.3.4. The list of parameters that were finally fed as input to the prediction model was obtained after applying feature selection methods. These methods can only be applied in scenarios where data centralization is feasible in an offline context. The 0 and 1 class represents the status of hospitalization rate related to asthma as Low and High.

However, a centralized architecture that signifies complete centralization from all the sites consequently involves enormous resources in terms of communication and computation. Therefore, an attempt was made to generate the prediction model based on the decentralized architecture. A heterogeneously distributed scenario, as stated before, consists of distributed sites where each has a different schema, and sites are unfamiliar with each others' data. Under the decentralized architecture execution, a distributed ensemble was implemented in the first instance, where each site builds its local model.

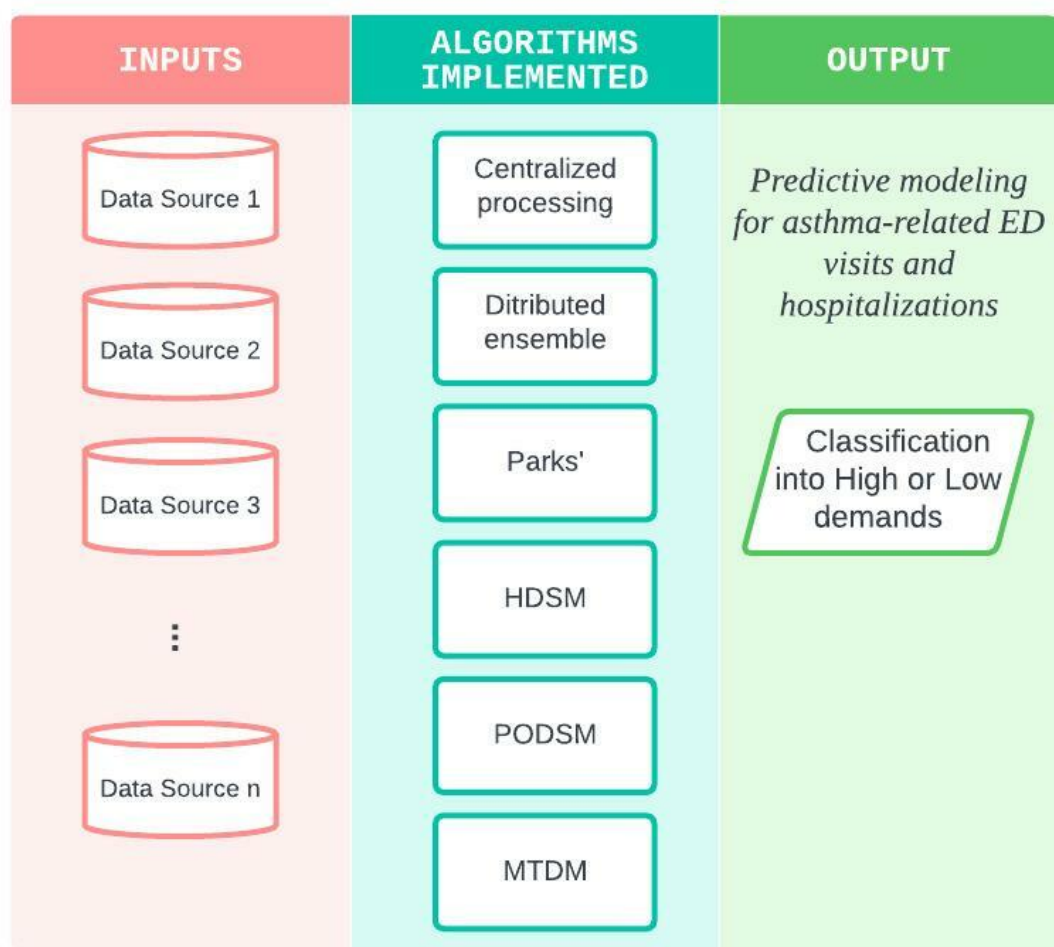


Figure 6.1: Pictorial representation stating the methodologies that would be applied to the asthma case study.

The final prediction is generated by combining the outcomes from distributed sites, using different aggregation mechanisms, maximum confidence, simple voting, and stacking. The subsequent execution was done using Parks' method and HDSM, which suggest the creation of trouble site or sites for learning cross-terms between distributed sites. All of these methods have been discussed in detail in Section 2.6. For executing all of these decentralized methodologies, 17 features were used because feature selection methods that were used in Section 5.4 by centralizing all the data in the offline mode were not applicable in this scenario. These 17 features were distributed into 12 primary sites, and features originating from the same location were allocated to a single primary site.

The output variable contains either 0 or 1 class representing the status of hospitalization rate related to asthma as Low and High. Online processing focuses on real-time analysis and continuous learning from streaming data, unlike batch processing, where the model is trained on a fixed dataset offline. Instead of retraining the entire model, online processing typically employs incremental learning techniques that update the model incrementally as new data arrives (refer to Section 2.3 for more details). These techniques allow the model to adapt and learn from the streaming data without the need for full retraining. PODSM and MTDM have been executed in Prequential evaluation environments (Section 3.4.2) and have been devised with an ability to adapt to concept drift.

The visualization of the model for implementing PODSM and MTDM has been presented in Figure 6.2. This figure depicts the working for an incoming record. In this case, we have inputs from 12 data streams or distributed sites. Please note the number of primary sites is fixed, but the hierarchy and number of trouble sites won't be fixed and would depend upon the state of the structure while processing that record. The aggregator gets the classification results from all the sites, which are then merged according to the aggregation methods (defined in Section 3.3.3) to give the final predicted classification result. This classification result is then compared with true classification to measure accuracy, and also, true classification is communicated to all the sites for training and updating their models. After training, this record is discarded. The record transmission that occurs between primary sites and trouble sites gets calculated as communication or TT. For MTDM, this TT would also include additional communication for centralized tree generation.

Furthermore, the PODSM algorithm discussed in Section 3.3.1, which aims to reduce communication and computational cost without sacrificing classification accuracy, was also applied to the asthma dataset. As explained, Bayesian Inference was used to estimate the confidence status of peer-distributed sites, thereby saving resources and

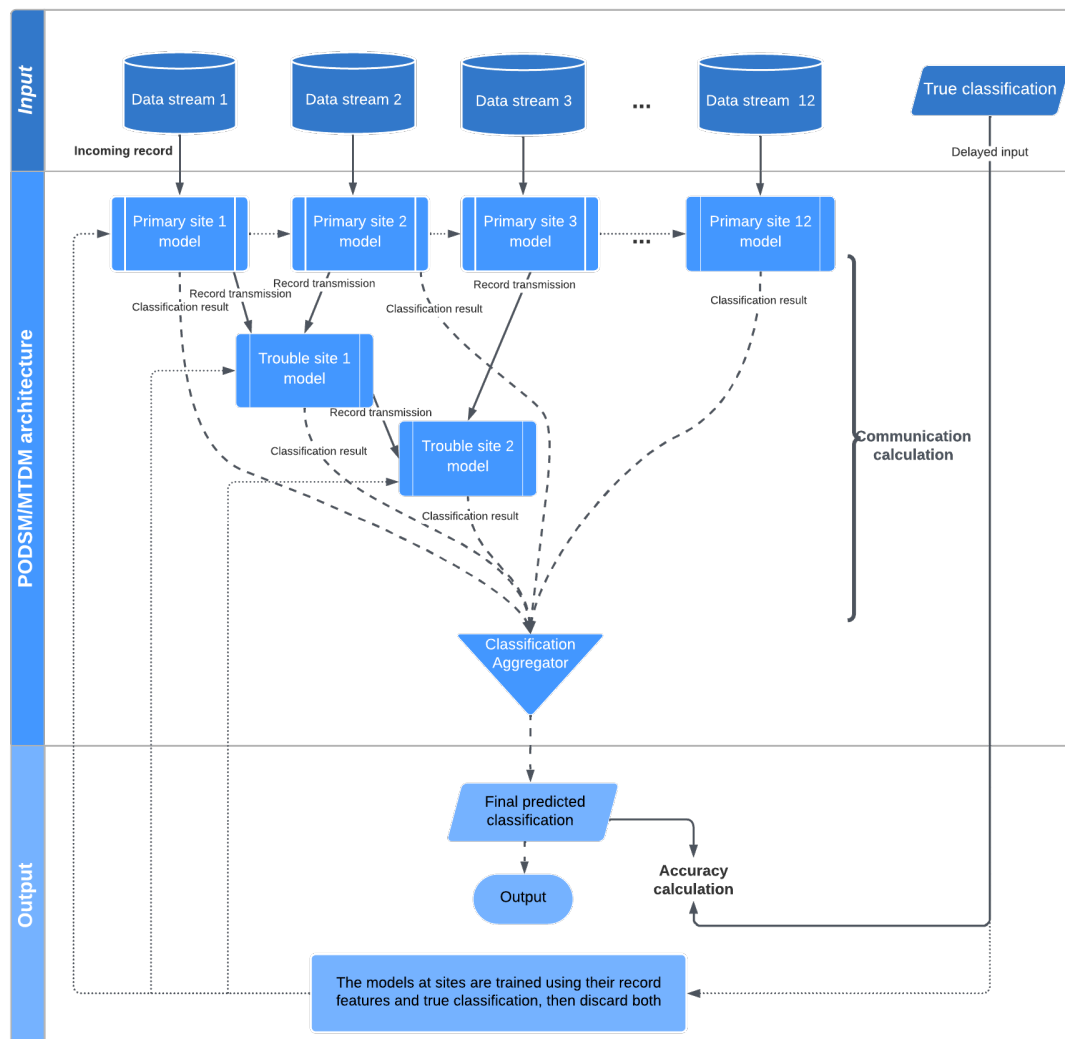


Figure 6.2: Pictorial representation stating the methodologies that would be applied to the asthma case study.

demonstrating its competency to maintain the trade-off between accuracy and resources utilized. Another algorithm, MTDM, presented in Section 4.3.1 was also implemented on the asthma dataset with the goal of efficiently learning the cross-terms between the distributed sites more efficiently. It uses minimized tree instead of a brute-force strategy (as used in HDSM) for exploring the relationship among the sites, thereupon witnessing significant improvement in resource time while maintaining the efficacy of classification.

The key challenge lies in collectively processing distributed data streams without

having any knowledge about their schema. Centralization helps to learn the cross-terms but involves extensive resource utilization. An endeavor to reduce resource utilization has an unfavorable effect on classification accuracy. Even though distributed ensemble calls for fewer resources, it lacks a mechanism to learn cross-terms, thus producing less accurate results. In a nutshell, this chapter focuses on testing the distributed processing algorithms with a central focus on PODSM and MTDM in online settings on the asthma dataset. It is targeted toward developing a prediction model that could predict days with peak demands for asthma-related ED visits and admissions using data from diverse heterogeneous distributed sources. The following section interprets the experimental evaluation of these algorithms:

6.4 Experimental Evaluation

The experiments for validating PODSM and MTDM on asthma dataset have been made available at two GitHub repositories with comparison to state-of-the-art, refer Appendix C.

6.4.1 Experiment configuration

All experiments were performed with OpenJDK 1.8.0 on a 64-bit Windows running on a 4×3.40 GHz Intel Core i5 CPU with 16 GB of memory. The configurations for executing the experiments were the same as PODSM and MTDM. The experiments were implemented using the Clojure programming language with the help of machine learning libraries from the MOA data stream mining system. As mentioned in Sections 3.4.1, all the models used Naïve Bayes classifiers at the primary sites, and ARF was the choice for the trouble sites. According to PODSM and MTDM, the aggregator site receives the classification results from all the sites and produces the final classification prediction using any aggregation methods mentioned in Section 3.3.3. The values assigned to the

set of configuration parameters that were kept constant throughout the running of all the decentralized experiments;

- Size-limit for all windows at sites and monitors - 1000 latest records
- Hoeffding bound (δ) in agreement and accuracy thresholds - 10^{-3}
- Threshold time period for creation and removal monitors - 500 records
- Smoothing factor for agreement threshold (θ) - 5
- Site utilization threshold for removal-monitors - 5%
- TF - 1x, 1.5x, 2x

For more details on the parameters, refer results section of Section 3.4.1.

6.4.2 Performance assessment metrics

A number of performance evaluation metrics were used to compare the performance of various algorithms that were used for predictive modeling for asthma using decentralized online techniques. A set of common metrics used for assessing the performance of both algorithms were - accuracy, TT, MTCP, and MTBR. The explanation of all these metrics has been elaborated in Section 3.4.2. As discussed in Section 3.3.3, there can be different ways of aggregating the classification results from all the sites. We will use maximum confidence, simple voting, and stacking to report the results.

6.4.3 Performance evaluation

Distributed ensemble, Parks', HDSM, PODSM, and MTDM were the experiments executed on the asthma dataset to evaluate the performance achieved by implementing different decentralized architectures in online settings. Above that, a distinction has been

drawn between decentralized and centralized architectures in the streaming context. For drawing performance comparisons between centralized and decentralized approaches using similar platforms and techniques, the experiments were also configured for a centralized experiment. An ARF classifier was trained for a centralized experiment containing all features with all the records portraying the entire data as centralized. With the basic features of RF, the adaptive part of ARF adds the capability to adapt to different types of concept drifts. The three main properties of an ARF algorithm are: *i)* it comprises an ensemble of decision trees that brings in diverseness through re-sampling using bagging, *ii)* an additional level of diversity is obtained by choosing a random subset of features for each splitting node, *iii)* each base tree is equipped with a drift and warning detector, which allows selective resetting to acknowledge an occurrence of a drift; when a warning is detected, a background tree training starts and replaces an active tree when this warning converts into a drift. To have an optimal number of records for calculating and evaluating the results using the defined metrics in the streaming context and also to allow decentralized structures to learn the cross-terms, the dataset for asthma was over-sampled to obtain a larger dataset because the original form contained a minimal number of records.

The experimental evaluation phase was further divided into two sub-parts; the first depicts the behavior of PODSM, while the second focuses on the execution of MTDM in the asthma case study. These sub-parts also comprehend comparison with state-of-the-art methodology. Both these algorithms, PODSM and MTDM, support a flexible hierarchy of trouble sites with the ability to detect drifts. The section underneath interprets these sub-parts in detail;

- **Implementation of PODSM** Apart from the list of parameters discussed above, an additional factor was kept constant, which was the size of *batch-1*; the default value assigned to *batch-1* was 15000. Table 6.1 analyses the results of

Table 6.1: Evaluation of state-of-the-art methods.

Metrics	Cent.	Distributed Ensemble			Parks'
	M.Conf	M.Conf	Vot	Stack	M.Conf
Accu	95.92	79.17	78.94	79.19	79.17
TT	1200	-	-	-	139.37
MTCP	1.35E5	1.01E4	8.14E3	1.17E5	1.20E4
MTBR	1.35E5	5.57E3	5.01E3	1.08E5	6.82E3

experimental implementation of state-of-the-art methods, namely, centralized architecture, distributed ensemble, and Parks' method in terms of accuracy, TT, MTCP, and MTBR. It can be seen that a centralized experiment has the potential to attain the highest accuracy. However, on another side, it involves an immense resource cost as it centralizes the complete dataset from all the distributed sites for generating classification results. Distributed ensembles generated using all three aggregation methods, although involve no data transmission¹ but witness a sharp decline in accuracy. The low accuracy is due to the lack of any technique to learn the cross-terms between distributed sites. Parks' method involves a fraction of communication which is considerably less than a centralized configuration.

Furthermore, HDSM polishes up the mechanism of learning the cross-terms, thus achieving a boost in accuracy. PODSM algorithm, dedicated to optimizing the usage of resources, tries to eliminate the unnecessary resources involved in Parks' method and HDSM. Table 6.2 shows the comparison drawn between HDSM and PODSM for all values of TF. The results in this table show a nearly 18% decrease in communication cost is gained while exhibiting a slight decline in accuracy for the maximum confidence aggregation method.

Additionally, Table 6.2 shows that PODSM is capable of achieving savings in terms of resource time as well, thus proving its competency in realizing savings

¹However, it involves the transmission of classification results from sites.

Table 6.2: Comparison between HDSM and PODSM for different aggregation methods.

Metric	HDSM			PODSM		
	M.Conf	Vot	Stack	M.Conf	Vot	Stack
Acc	87.65	89.17	88.92	83.99	84.97	84.81
TT	285.62	367.34	368.90	234.96	298.06	300.32
MTCP	1.19E5	1.01E5	2.04E5	9.98E4	9.90E4	1.93E5
MTBR	4.48E4	4.31E4	1.43E5	3.77E4	3.30E4	1.37E5

in resources while maintaining similar accuracy levels. The results in Table 6.2 revealed that maximum confidence achieves maximum gains among the three methods while comparing HDSM and PODSM. The timeline for accuracy, data transmission volume, and resource time are displayed in Figures 6.3 and 6.4. The first timeline in both graphs depicts the accuracy plots, with the sequence of addition and removal of trouble sites, shown as green and red vertical lines, respectively. The second plot shows the graphs for data transmission volume while the third and the last plots are dedicated to resource time utilization. It can be witnessed by applying algorithm PODSM; after *batch-1*, the transmission volume and resource time reduce when the calculated statistics become operative.

The size of *batch-1* was 15000 records for these representations, and until this stage, all the records were used for calculating statistics or feedback values. Thereupon, these statistics or feedback values were applied to the remaining set of records, which is small in this case. Therefore, to depict a clear picture of PODSM algorithm evaluation, the dataset was under-sampled and over-sampled to reveal the application of feedback statistics for both cases. Figures 6.5a and 6.5b give a glimpse of these scenarios, demonstrating the behavior of PODSM for different sizes of data. The region under red boxes reflects how the statistics calculated using *batch-1* affect the transmission volume for the remaining records. Both the cases shown in Figure 6.5 demonstrate the usage of statistics on a more significant number of records. The above plots in Figures 6.5a and 6.5b represent

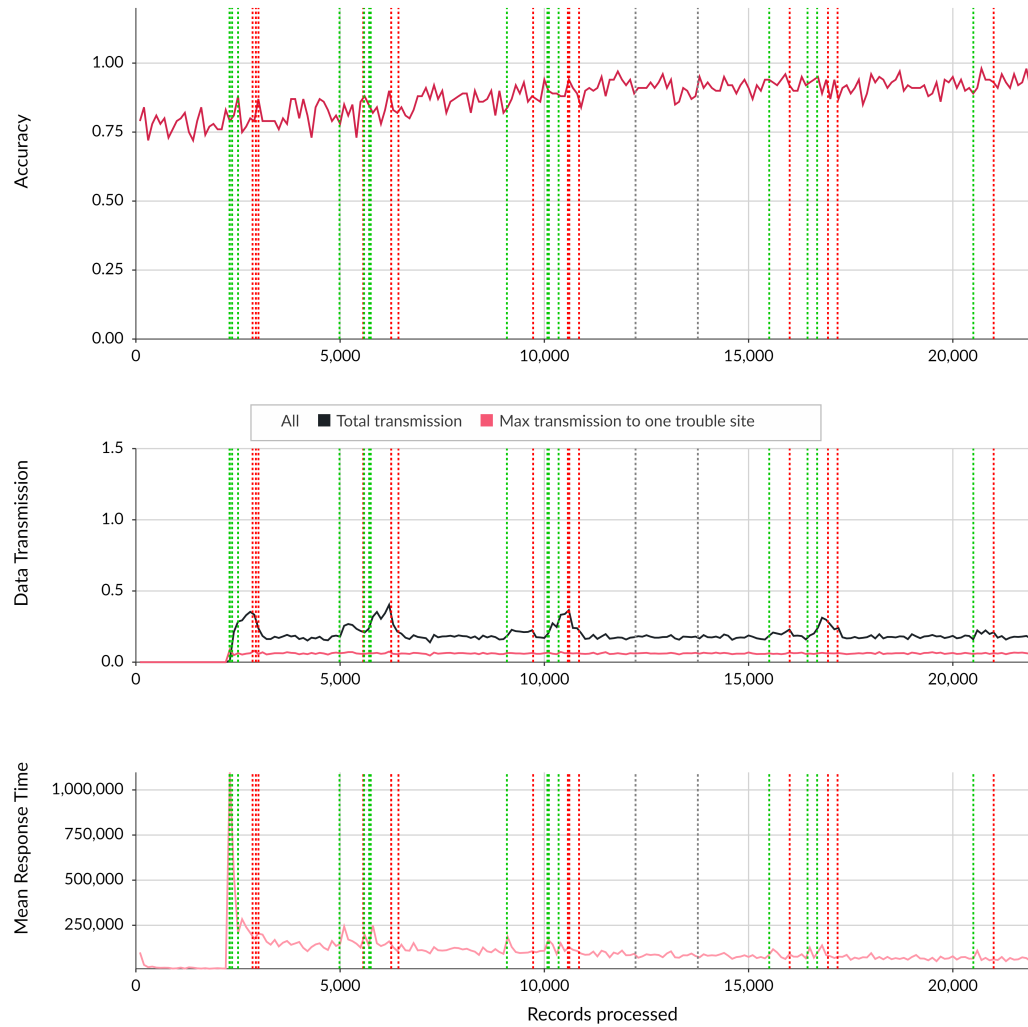


Figure 6.3: Timeline for HDSM

the timeline for HDSM, while the lower one presents the PODSM. On applying PODSM, some increase can be seen in communication during the creation of a new trouble site which involves the transmission of records for calculating statistics.

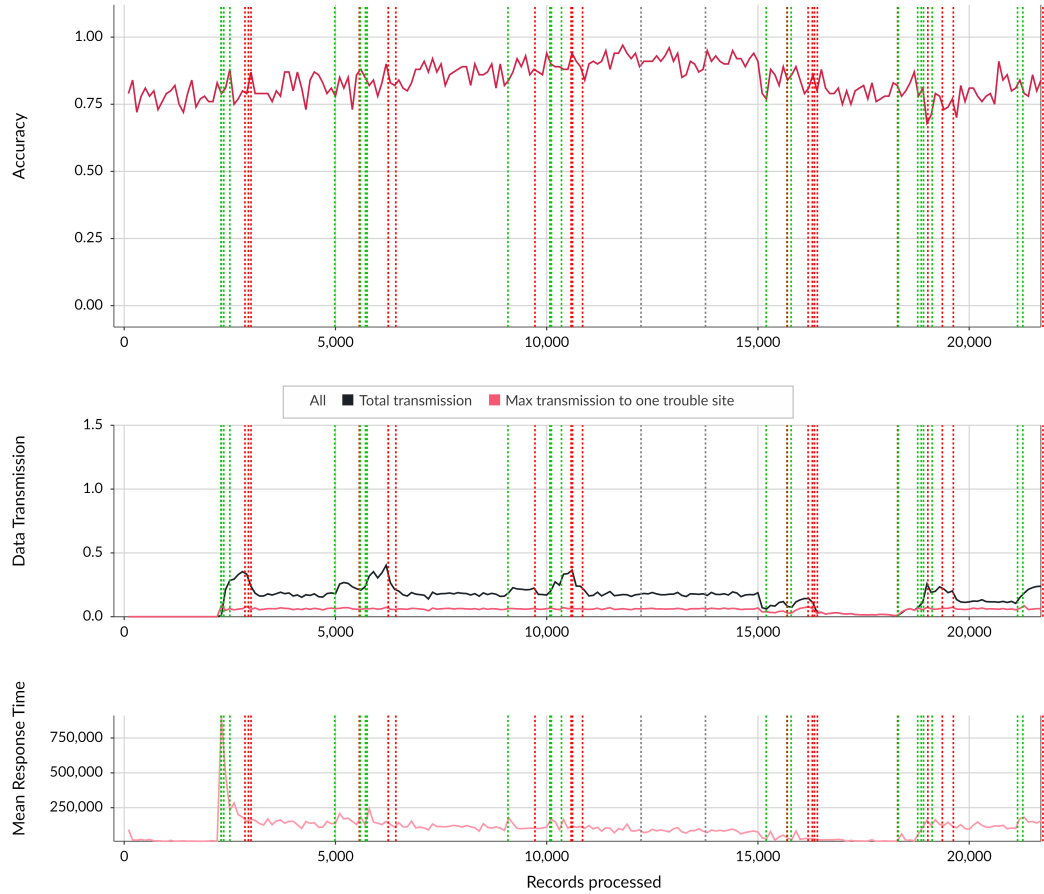


Figure 6.4: Timeline for PODSM.

Parameter sensitivity analysis of PODSM

The subsequent implementation phase focuses on checking the parameter sensitivity analysis of PODSM. This algorithm includes two parameters that can be tuned for optimization as discussed in Section 3.4.4; these were - i) size of *batch-1* and ii) to change the frequency of updating the statistics/feedback from the trouble site to the source sites.

- **Changing size of *batch-1*** The standard value of *batch-1* used throughout the experimentation phase was 15000. For checking the sensitivity of the algorithm based on the size of *batch-1*, it was assigned two different values



Figure 6.5: PODSM's impact on communication for different sizes of the dataset.

of 10000 and 17000 in response to decreasing and increasing the size of *batch-1*. Figure 6.6 shows the interpretation of accuracy and TT wrapping

all the three aggregation methods when the size of *batch-1* was 10000. The point values for accuracy and total transmission were overlapping for MaxConf and Voting methods at TF 1.5x according to their respective scales, and thus, the red line is hiding behind the blue color line. Under this *batch-1* size, maximum accuracy is achieved at TF 1x under the simple voting aggregation method while saving communication costs of approximately 34% when compared to HDSM with a similar configuration. When stacking is used as the aggregation method, at TF 1x, approximately 32% of earnings are achieved for communication.

When the size of *batch-1* changes to 17000, as shown in Figure 6.7, communication cost benefits are realized for all the three methods, with maximum gain portrayed in the stacking aggregation method when TF equals 1x. The savings in terms of communication are less when the size of *batch-1* is 17,000 as compared to 10,000. This is because at *batch-1* size equal to 17000, even though the statistics calculation has been done on more number of records but the remaining set of records when these statistics or feedback values are applied is very small.

Furthermore, resource time balancing is shown in Table 6.3. These results were demonstrated for all the aggregation methods plus all configurations of TF. It can be observed that the size of *batch-1* equal to 10000 improves the computation time, and more precisely, a major improvement is observed for MTBR.

- **Changing the frequency of updating statistics** The other parameter for which the PODSM algorithm's sensitivity was analyzed specifies the frequency with which the statistics or feedback are updated. The statistics values can be updated at the trouble site either on a record basis (on the

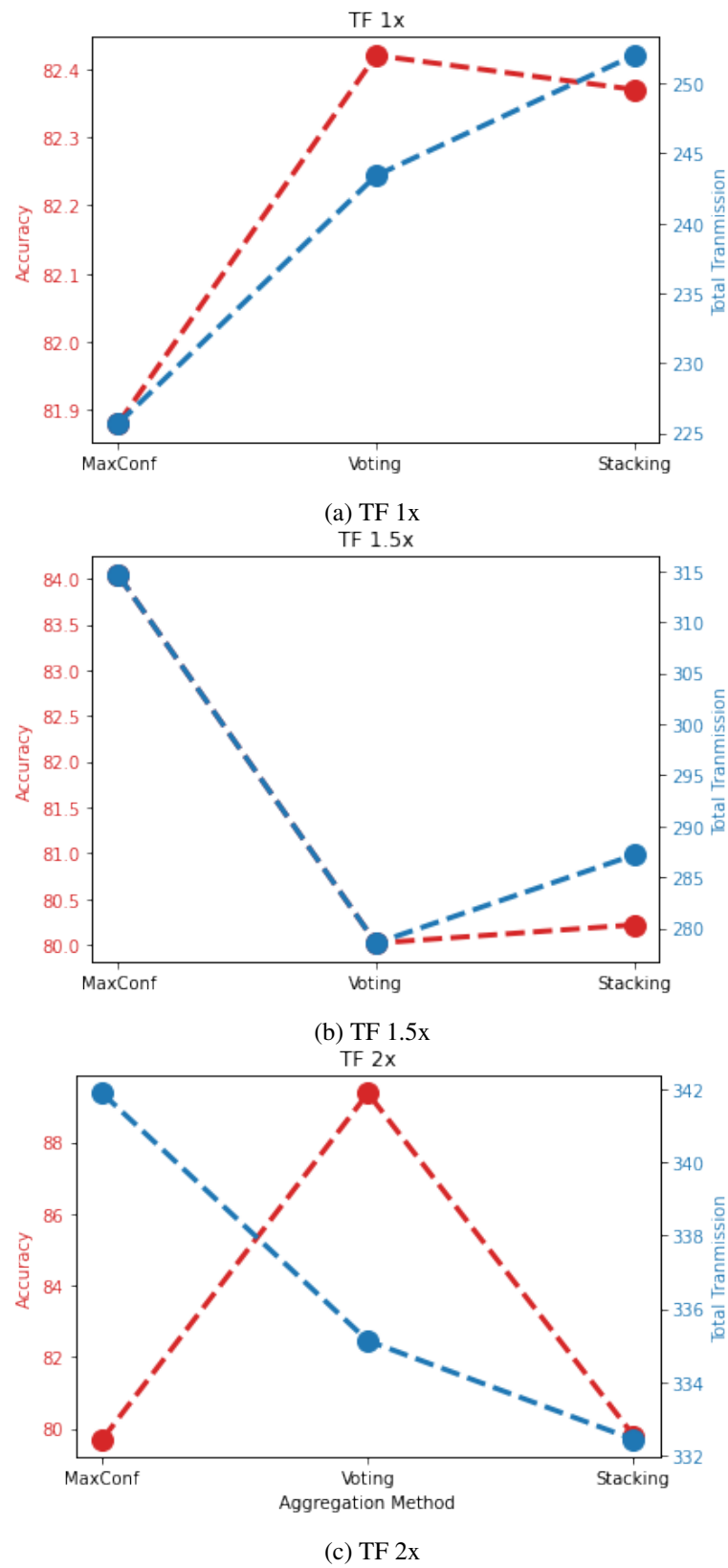


Figure 6.6: Accuracy and communication evaluation when the size of *batch-1* is 10000.

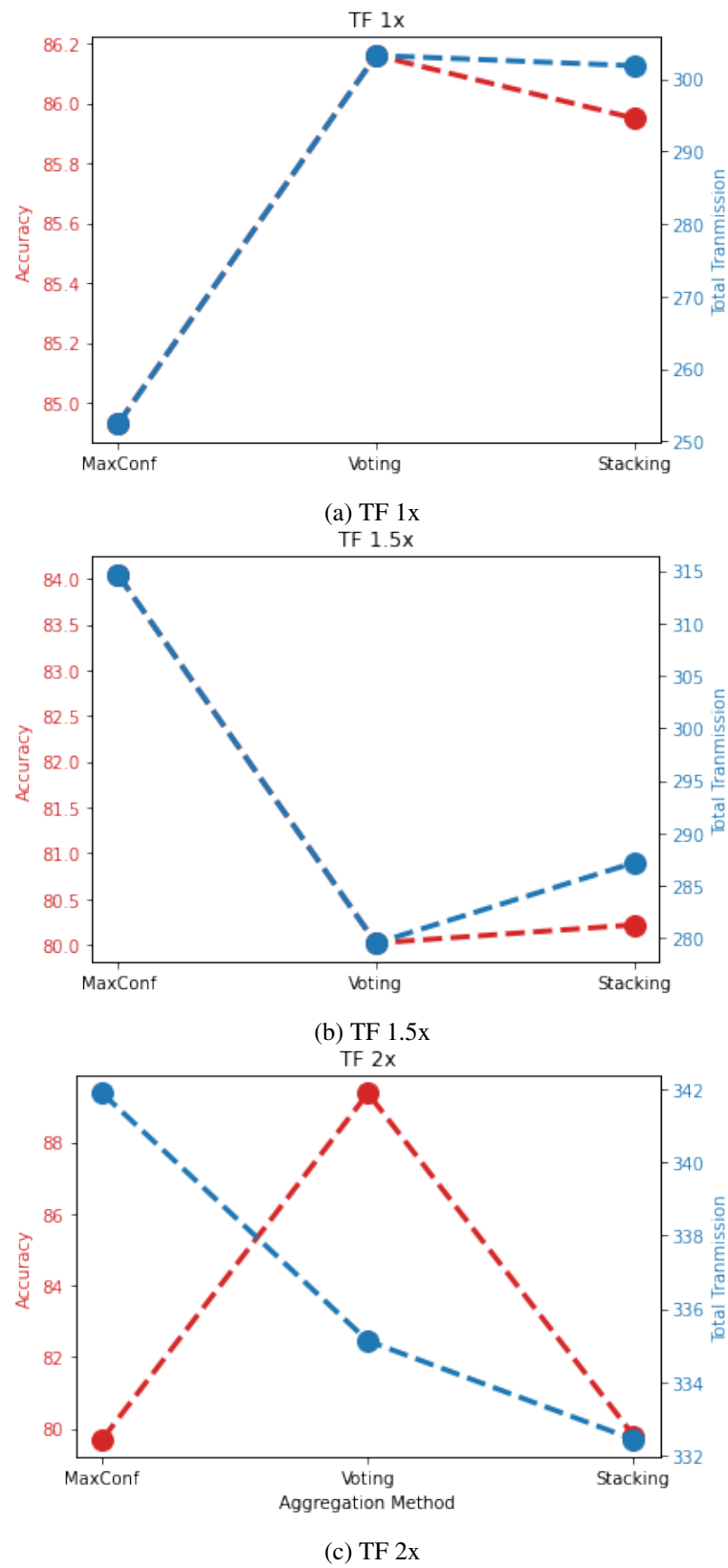


Figure 6.7: Effect of *batch-1* size attaining a value of 17000 on accuracy and communication.

Table 6.3: *Batch-1* sensitivity analysis in terms of MTCP and MTBR.

Experiment	<i>batch-1</i> 10000		<i>batch-1</i> 17000	
	MTCP	MTBR	MTCP	MTBR
M.Conf, TF 1x	9.90E4	2.72E3	1.15E5	4.65E4
Vot, TF 1x	9.40E4	2.28E3	9.65E4	3.85E4
Stack, TF 1x	1.98E5	1.33E5	2.01E5	1.43E5
M.Conf, TF 1.5x	1.49E5	5.11E4	1.47E5	5.15E4
Vot, TF 1.5x	1.18E5	1.52E4	1.22E5	1.58E4
Stack, TF 1.5x	2.10E5	1.48E5	2.15E5	1.51E5
M.Conf, TF 2x	1.46E5	1.95E4	1.51E5	2.18E4
Vot, TF 2x	1.47E5	8.11E4	1.47E5	8.15E4
Stack, TF 2x	2.27E5	1.54E5	2.30E5	1.57E5

arrival of each record) or on a change in the hierarchy of trouble sites (when some trouble site is created or removed). For checking this parameter, the frequency of updating the statistics was changed from a record basis to the case when a change in hierarchy triggers the updation of statistics or feedback values. The *batch-1* size used for this analysis was 15000. Even though the statistics were not updated regularly on a record basis, Table 6.4 shows the competency of PODSM in producing efficient results compared to state-of-the-art methods. As such, this version of PODSM reduces communication by nearly 15% while balancing out accuracy quite well for the maximum aggregation method at TF 1x. Further, savings in resource time can be spotted in Table 6.4b when compared to recent methods presented in Tables 6.1 and 6.2.

Figure 6.8 shows the ranking comparison for accuracy and communication between the centralized experiment and three versions of PODSM when TF equals 1x, 1.5x, and 2x for the maximum confidence aggregation method. The better ranks are represented by points near to origin. It can be seen that TF 1.5x was ranked as the best configuration with the ability to maintain a good trade-off

Table 6.4: Sensitivity analysis of PODSM on the asthma case study by changing the frequency of updating statistics.

Trouble Factor	Accuracy			TT		
	M.Conf	Vot	Stack	M.Conf	Vot	Stack
TF 1x	84.74	85.97	85.62	243.76	298.74	297.67
TF 1.5x	84.04	80.02	80.22	314.61	279.51	287.17
TF 2x	79.70	89.37	79.81	341.89	335.13	332.44

(a) Accuracy and communication metrics

Trouble Factor	MTCP			MTBR		
	M.Conf	Vot	Stack	M.Conf	Vot	Stack
TF 1x	1.01E5	1.00E5	1.95E5	4.20E4	3.78E4	1.38E5
TF 1.5x	1.43E5	1.22E5	2.45E5	5.29E4	1.58E4	1.67E5
TF 2x	1.67E5	1.66E5	2.48E5	2.30E4	9.45E4	1.71E5

(b) Resource time in terms of MTCP and MTBR

between accuracy and communication. The root cause behind this behavior is that at TF 1.5x, the score was best for accuracy, entailing that level of communication.

- **Implementation of MTDM** For implementing MTDM, apart from setting the standard parameters, additional parameters were kept fixed during the execution. The three additional parameters used in this case were grace period, split confidence, and minimum records. The explanation of all these parameters has been discussed in Section 4.4.1. The default value of parameters used for implementing MTDM on the asthma dataset was:

- Grace Period 5
- Split Confidence 0.9
- Min records 150

Three additional metrics were used for gauging the performance of MTDM with their details previously mentioned in Section 4.4.2; they were TC, TCM, and

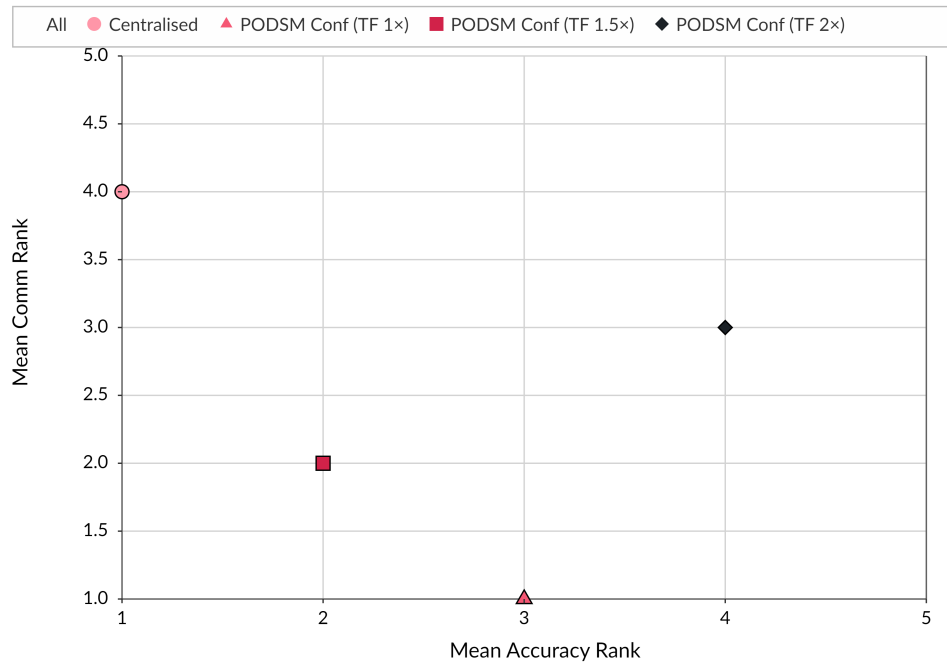


Figure 6.8: Accuracy and communication ranking between centralized and PODSM.

Table 6.5: Results after implementation of MTDM.

Acc	TT	MTCP	MTBR
88.31	302.58	9.46E4	2.56E4

TFTS. Table 6.5 presents the results obtained after implementing MTDM for the maximum confidence aggregation method. The outcome can be compared to the results depicted in Tables 6.1 and 6.2 for state-of-the-art work (Distributed Ensembles, Parks', and HDSM). MTDM's aim to learn the relationships between distributed sites brought favorable results for resource time and accuracy but added a little communication cost compared to HDSM.

The ultimate goal behind MTDM was to decrease the number of combinations explored for pairing primary sites by learning their behavior in terms of cross-terms, which ultimately reduced the resource time utilization. As the ideology of using a hierarchy of trouble sites and estimating the best subsets of distributed sites

Table 6.6: Comparison between HDSM and MTDM in terms of TC, TCM, and TFTS.

Metrics	HDSM			MTDM		
	M. Conf	Vot	Stack	M. Conf	Vot	Stack
TC	66	66	66	20	20	20
TCM	0.38	0.14	0.16	0.27	0.10	0.09
TFTS	4600.03	4523.65	4927.44	1482.46	1304.83	1416.84

are used only in HDSM and MTDM, hence the other metrics TC, TCM, and TFTS were evaluated only for these two methods. Table 6.6 shows the comparative study among these two methods using these metrics for all aggregation methods with TF set at 1x.

It can be noticed that the number of combinations explored remains the same within the same algorithm for all aggregation methods. Under HDSM, the number of pairs comes to be 66 using the combinations formula $C(12, 2)$, whereas by executing the tree-based algorithm MTDM, the number of combinations explored decreases to 20. Moreover, the resource time utilized for initializing creation monitors and the time taken to generate the first trouble site has also been turned down. The results presented in Table 6.5 and 6.6 show that the execution of MTDM on the asthma dataset has included a minor communication cost which brings a huge reward in terms of resource time and accuracy. The Figures in 6.9 and 6.10 display the visual distinction on implementation of HDSM and MTDM. The comparability has been drawn in terms of accuracy, data transmission volume, and resource time. For HDSM, Figure 6.9 witnesses a sharp inclination at the initialization phase due to the brute force strategy used to find optimal pairs of sites, which involves huge resource time utilization for exploring and keeping track of a large set of combinations.

Another metric used to compare HDSM and MTDM was TUCM which expresses the summary of the time taken to update the agreement between the creation

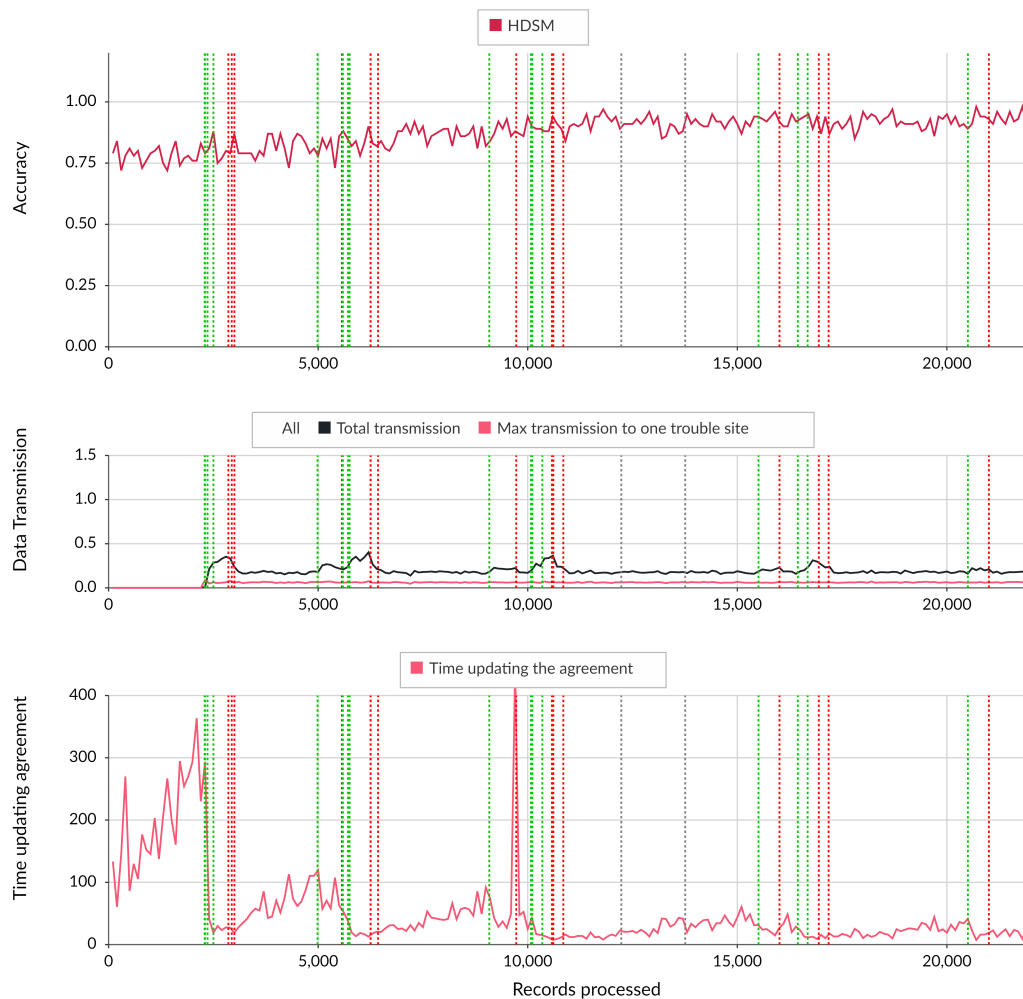


Figure 6.9: Timeline for HDSM for comparing it with MTDM.

monitors maintained for each combination of sites throughout the experiment. The central focus of MTDM has been reducing the pairing options explored, which ultimately positively impacted the resource time needed to update the agreement. Figure 6.11 gives the pictorial representation of the comparison between HDSM and MTDM in terms of this resource time usage. It can be observed that for all the aggregation methods plus for all configurations of TF, MTDM verifies its competency in cutting down the resource time and overload that HDSM induces by maintaining a large number of creation monitors for each combination. Another point to note is that the gap between HDSM and

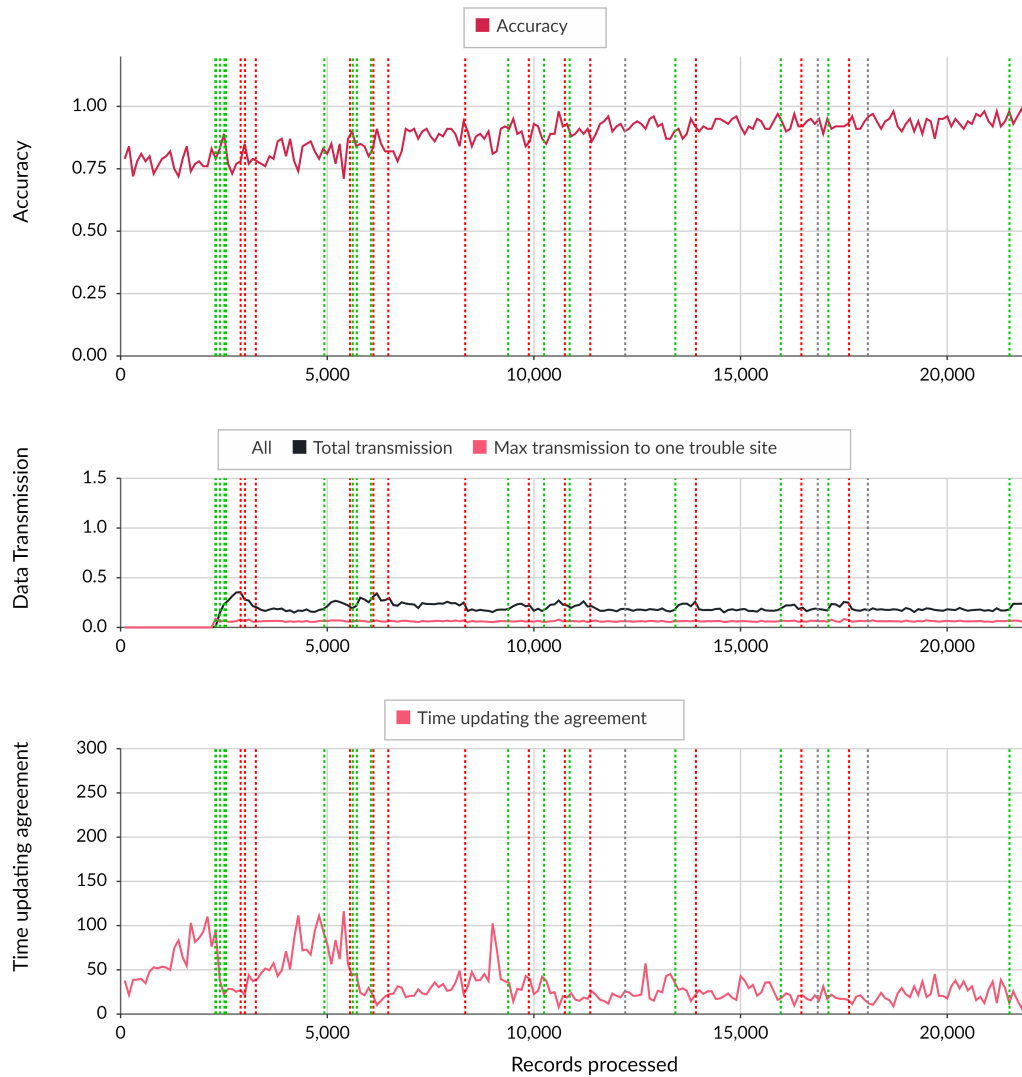
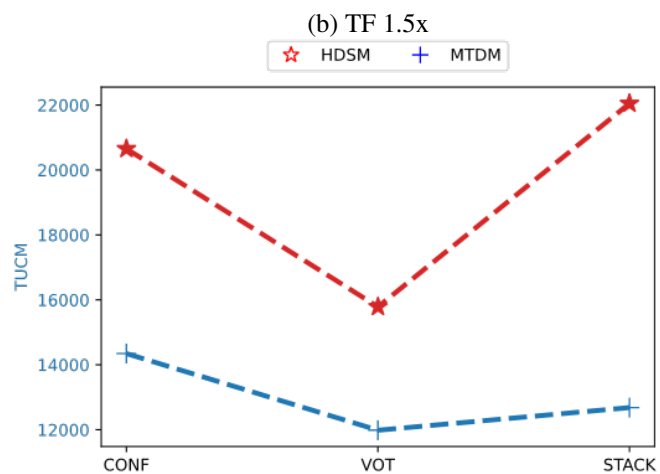
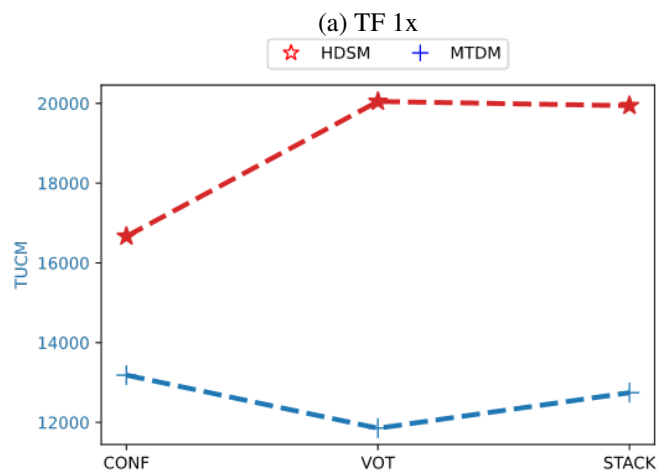
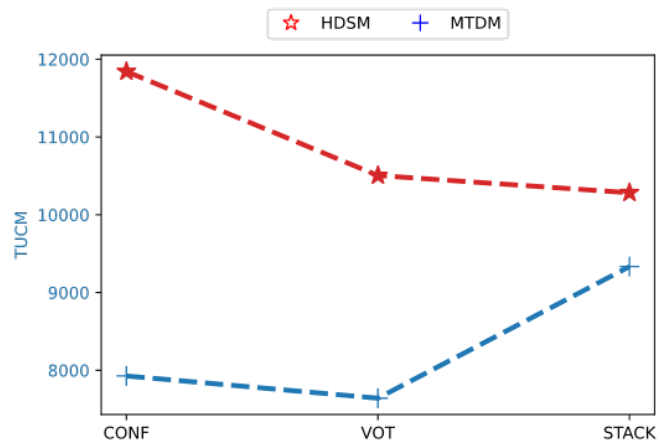


Figure 6.10: Timeline for MTDM.

MTDM graphs for all the methods becomes wider at a higher value of TF in Figure 6.11. The savings in resources can be realized by applying MTDM under any aggregation method, as its applicability brings fruitful results for all cases.

Parameter sensitivity analysis of MTDM

Next, we test the sensitivity of MTDM depending upon its customized parameters, that is, grace period, split confidence, and minimum records. These parameters



(c) TF 2x

Figure 6.11: Difference between HDSM and MTDM in terms of TUCM.

Table 6.7: Impact of changing values of sensitive parameters.

Parameters	Value	Acc	TT	TC
Grace Period	10			
Split Conf	0.9	87.32	326.08	46
Min records	100			
Grace Period	5			
Split Conf	0.9	88.61	327.67	13
Min records	200			
Grace Period	1			
Split Conf	0.9	86.69	272.03	21
Min records	150			
Grace Period	5			
Split Conf	0.7	88.25	364.23	38
Min records	200			

were assigned different values, and the results are demonstrated in Table 6.7. We can see that for the second case (when grace period = 5, split confidence = 0.9, and minimum records = 200), even though the value for TC further reduces to 13 and accuracy also holds a comparable value, it includes more communication when compared to other cases listed in Table 6.5. Similarly, for other cases also it is noticeable how changing the value of these parameters may induce modified values for metrics, namely accuracy, TT, and TC.

As discussed earlier, the working of MTDM depends upon a minimized tree generated using a minimum set of records from all the distributed sites. Allowing a different number of records to generate a minimized tree and adjusting other parameters leads to differing results for explored combinations and the resource time required to generate and parse this minimized tree. The effect of using a different number of records to generate the minimized tree has been shown in Figure 6.12. The first graph portraying the number of combinations, observes a sharp declination with the increasing records. At the same time, a small utilization

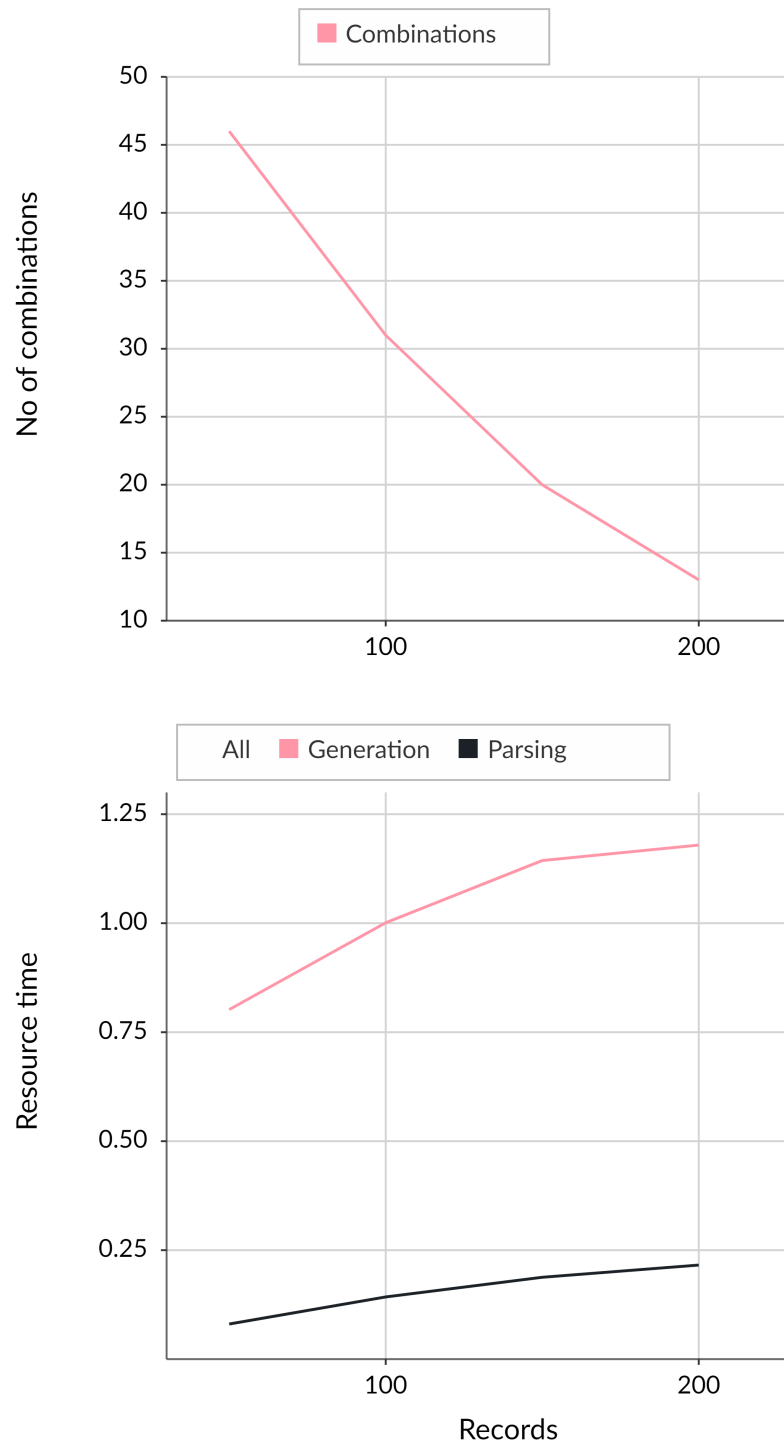


Figure 6.12: Footprint of the number of records on TC and resource time for processing a tree.

of resource time is also added for generating and parsing a tree which ultimately earns massive economization in efficiency.

6.5 Applicability and Limitations

We have demonstrated the prominence of exploiting diverse sources for predicting peak demand days for asthma in near real-time. As such, this data-driven approach can provide early warning signals to healthcare providers and administrators for adverse events such as overcrowding, enabling timely, proactive, and targeted preventive and therapeutic interventions. Implementation of such techniques can serve as a demand prediction tool, triggering demand and capacity management action. The outputs achieved from such models may help in avoiding issues like ED overcrowding, which is a global concern and can ensure qualitative and non-disruptive healthcare to asthma patients when required. A cost-effective decentralized architecture implemented with geographic stratification can provide a more concise prediction according to area segmentation. Such a model can also be applied to other diseases like COPD and Cardio-Vascular diseases. These predictive modeling techniques can provide probable productivity to healthcare providers and organizations in predicting events before they occur, allowing early medical intervention, even according to geographic segmentation.

Concerning the dataset, more data sources can be investigated to check their correlation with asthma, and specifically, other social media platforms like Twitter and news blogs can also be examined. Being limited to a single consolidated geographic area, this study lacks implementation of predictive modeling according to geographic segmentation. A model implemented with geographic stratification can predict peak demands according to geographic locations, enabling more precise and concise proactive measures and actions to adapt and manage resources according to changes in demand. In the current implementation of PODSM and MTDM, there lacks a technique for

preprocessing the data values originating from heterogeneous distributed data streams, so the data streams observed at primary sites are presumed to be preprocessed. In a practical scenario, data preprocessing steps can be performed at the primary sites before data is fed to the model.

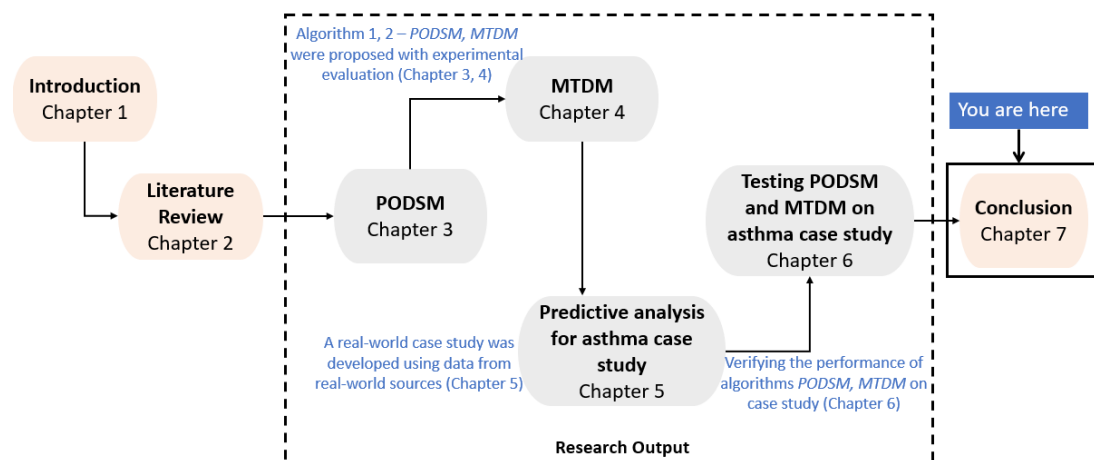
6.6 Chapter Summary

This chapter presented a unique perspective on community health management, particularly for asthma, a crucial health issue in NZ. In this work, we exploited various distributed heterogeneous data sources and demonstrated their potential to build resource-optimized modeling for predicting peak demands for asthma-related ED visits and hospitalizations. Work has been done on the transition from centralized to decentralized modeling, which eliminates the need for complete data fusion at a single site and proves an efficient structure for prediction modeling. An initiative has been undertaken towards online processing of heterogeneous streams, which can predict the hospital demands for asthma in near real-time. Further work was done towards making this prediction model economical by bringing down communication and computational costs without compromising accuracy. The decentralized methods (PODSM and MTDM) discussed in this research work demonstrated their capability and performance towards conjoint processing of heterogeneous distributed data streams in any resource-constrained scenario. Moreover, the algorithm's capability to maintain a balance between accuracy, communication, and resource time makes them flexible enough for a diverse range of applications.

Chapter 7

Conclusion

This chapter wraps up the complete thesis by redefining the research aims targeted in this study via a suite of algorithms proposed after identifying research gaps through the literature review. A subsequent discussion has been provided on variations that can be applied to these algorithms. Accelerating further would provide details on the restrictions that may exist in the applicability of the solutions proposed in the form of an algorithm for each research question. Ultimately, the future scope of this work has been suggested by highlighting interesting extensions and potential directions.



Roadmap of the thesis.

7.1 Research Achievements

This research focused on vertically distributed data streams which may also be in the form of heterogeneous data sources. We presented a suite of novel algorithms that can be applied for mining data in a resource-restricted environment, where the data is heterogeneously distributed across geographic sites. The work was done targeting the maximization of throughput in the stream classifications context while keeping in mind the constraints attached. Expressly, it tries to maintain a balance between accuracy, communication, and resource time. Two different resource constraint scenarios were identified in Section 2.7, and as a result, two algorithms, PODSM (Chapter 3) and MTDM (Chapter 4) were presented as solutions for first two research questions. The third phase focused on uncovering a case study with real data from real-world sources for testing the practical implications of the proposed algorithms PODSM and MTDM. The case study was focused on developing a model that could predict peak demand days for asthma-related ED visits and hospitalizations, discussed in Chapters 5 and 6. In the former chapter, the stages involved in building this real-life case study have been examined in detail, whereas the latter focuses on validating PODSM and MTDM in this case study. A description illustrating the competency of the methodologies applied in this thesis, work towards attaining their goals, as underlined below:

1. **Devising a novel algorithm for optimizing performance, *PODSM***- The main feature of this algorithm was to reduce the cost spent on resources for the classification process while maintaining accuracy levels. Hence, this class of algorithms was used for scenarios with restrictions imposed on communication or resource time costs. For devising a method for such a case, an underlying hierarchical structure was employed for classifying vertically distributed or heterogeneous data sources. This structure considered the pairing of vertically distributed sites or heterogeneous sources into subsets of two. The principal ideology used towards

optimizing resources was to decrease communication flow, which was wasted and served no purpose. A statistical solution based on Bayes' theorem was adopted, and the theorem was customized to calculate the conditional probabilities to make them applicable in our problem domain. An unconfident site used the formulated version of this theorem to estimate the likelihood of paired sister site being confident for a record. This estimation helps an unconfident site to predict whether its paired site would be confident or unconfident without knowing its sister paired site's view (feature set or data). Depending upon the predictions, the unconfident site decides whether to send or skip sending the record to the trouble site. The experimentation evaluation was conducted to test the competency of PODSM on eleven benchmark datasets, revealing significant savings in resources regarding communication and resource time can be attained while achieving a similar accuracy level. *The best results were achieved for the HIGS dataset, which witnessed a drop of 34.66% and 27.06% in communication and resource time compared to the most recent method. Furthermore, instead of showing a detrimental effect on accuracy, it earned an increase of 0.44% in accuracy (refer Section 3.4.3 for details).*

This algorithm has introduced a mechanism that helps an unconfident to predict if its paired site would be confident/unconfident for a record. With this scrutinizing process, it becomes possible for an unconfident site to make a decision before sending its record to the trouble site. As a result, an additional intelligence feature is added to the sites, which thereby helps in reducing the communication and computation involved in the classification process. PODSM serves as a novel contribution to mining heterogeneously distributed data streams in an environment with restrictions on communication and computation.

2. Developing a scalable and efficient method for learning the relationship

between heterogeneous distributed data sources, MTDM- Although the heterogeneous data sources located at distributed locations may have different views, the feature set available at these sites differ, but these features may observe some direct or indirect relationship. The prime attraction of the algorithm is to learn this relationship efficiently without much overhead cost and with minimal possible communication, while preserving accuracy. The fundamental structure followed throughout this research work, suggests pairing the sites into pairs of two depending upon the level of agreement between the two sites. MTDM proposes generating a minimized tree and parsing this tree to gauge the level of agreement instead of using the brute-force strategy to hunt for the best pairs of distributed sites. This methodology, in turn, brings a cutback in resource time utilized in maintaining the magnitude of agreement between all possible combinations. This algorithm was assessed on eleven standard machine learning datasets, and for all of them, high-yielding results were obtained for resource time and the number of combinations explored to learn the relationship between the sites. Along with this, an elevation in accuracy was also spotted for most of the datasets. *For one of the datasets, FOG, MTDM achieved 37.65% improvement in resource time (TFTS¹), combination explored decreased from 36 to 22 and showed accuracy improvement of 1.33% when compared to HDSM, but added a communication cost of 11%. MTDM attained an accuracy improvement of nearly 51% when compared to Distributed ensemble (refer Section 4.4.3 for details).*

MTDM provides an efficient and scalable procedure to learn the cross-terms between the distributed sites. The brute force strategy used in existing studies overloads the system by checking all the possible combinations of distributed sites. Through MTDM, we obtain a lightweight solution to capture the cross-terms and get the optimal pairs for generating trouble sites. This solution is scalable in

¹A metric used for comparing resource time in MTDM

the sense that it will not collapse even if the number of sites becomes enormous. Moreover, it shows an immense improvement in the resource time consumed in searching the optimal pairs of sites for forming the trouble sites.

3. **Building up a real-world case study and testing the applicability of PODSM and MTDM on this case study-** As demonstrated, the algorithms PODSM and MTDM were designed to acknowledge research gaps spotted while mining heterogeneous distributed data streams. A case study was developed using real-world data to check the applicability of the algorithms, PODSM and MTDM, and verify their potential in a real-world scenario. The basic ideology behind this phase of the study was to monitor the behavior of these proposed algorithms and check what impact they would probably cause on a real dataset. The case study chosen aimed at developing a forecasting tool for predicting peak demand days on which asthma-related emergency hospitalizations are greater than the threshold value, using data from heterogeneous sources, including meteorological factors, air quality parameters, and social media data. Thus, the data attributing to heterogeneous sources has been collected from different distributed sites spanning the geographic area of Auckland, NZ. The principle behind this problem was to classify the target variable expressing asthma-related emergency hospitalizations as Low or High using input from diverse sources mentioned above. In the first phase, the data was centralized from different sources and was processed using an offline approach. Several stages were involved in executing predictive modeling for asthma, including i) data collection, ii) data preparation, iii) feature selection iv) prediction model. For developing a prediction model, several classical machine algorithms were applied to generate an offline prediction model by centralizing all the data. In the subsequent phase, all the algorithms based on distributed processing architecture, distributed ensemble, parks' method, HDSM, PODSM,

and MTDM were implemented in this asthma case study in the streaming context. The experimental evaluation proved the proficiency of the proposed algorithms as part of this research work by achieving their goals in a live case study. *The best prediction model developed using standard machine learning algorithms by centralization in offline mode achieved an accuracy of nearly 91%. In the streaming context, the centralized model achieved an accuracy of approximately 96%, though the configuration of experiments was different for both. The best case of PODSM was capable of achieving an accuracy of nearly 85% involving 298.06% of transmission, whereas the accuracy level for MTDM was 89% with 302.59% of transmission. The combination explored decreased from 66 to 20 after the application of MTDM with 67.77% improvement in resource time in terms of TFTS (refer Sections 5.4 and 6.4.3 for details).*

This work has made a novel contribution towards predicting peak demand days for ED visits and asthma-related hospitalizations using data from heterogeneous sources. A decentralized architecture implementation offers an advantage over centralized processing regarding resources. The execution of novel algorithms, PODSM and MTDM, in this case, deliver an economical solution for implementing the predictive modeling according to geographic stratification.

7.2 Variations of Work Proposed in this Study

The algorithms proposed in this work, *PODSM and MTDM*, have demonstrated their competency on eleven benchmark datasets and also on a live case study. This section discusses the variations that can be applied to individual research solutions for further improvement in accuracy and resources or changing the configurability.

1. **PODSM:** The PODSM algorithm has proved its capability to attain efficiency

in terms of communication and resource time. In PODSM, an unconfident site calculates the probability of its paired site being confident, which was then compared to a static value of 0.50. The reason for choosing 0.50 was explained in Section 3.3.1. An alternative approach can be employed to estimate the confidence value, which attempts to approximate the confidence level of a paired site. In order to compute this, the formula stated in Equation 3.4 must be modified to include the confidence value of site B. For this, source sites must transmit their confidence level with the record fragment to the trouble site. Instead of maintaining statistics in the form that helps an unconfident site A to gauge whether its paired site B would be confident or unconfident for a particular record, another form of estimation can be made for predicting the confidence value of B for that record. Further, this estimated confidence level can be compared to site B's actual mean confidence level. Remember, the confidence levels of paired source sites are not communicated to each other but are shared with the trouble site, which calculates the statistics and transmits them back to the source site as was done in PODSM.

2. **MTDM:** For MTDM, a minimized tree was generated to find the cross-terms between heterogeneous distributed sites. This algorithm was applied in the initialization phase to save the system from exploring a vast number of combinations for learning the cross-terms, ultimately decreasing the resource time. This algorithm used a tree to find the cross-terms at the initialization phase. The output from the tree provides the best pairs of distributed sites, decreasing the load of maintaining a large pool of information for each combination. By this, we successfully portrayed tree proficiency in searching for cross-terms. A modified form of this algorithm can be created, adding the capability to update the tree when a drift is detected. Consequently, this would require transmitting the recent minimum set of records from all the distributed sites to the centralized site to modify the tree

whenever there is concept drift. With this modified version in place, the overhead of keeping track of the level of agreement can be inhibited completely.

3. **Asthma case study:** The threshold used to generate categorization for asthma-related ED visits and hospitalizations by dividing the continuous hospital data into classes, can be modified. In our work, we used the 70th percentile to divide the data into two classes; Low and High. Likewise, another configuration can be generated by dividing the continuous hospital data into three classes: Low, Medium, and High, permitting the impact of making these alterations to be checked. Making these modifications would directly affect defining the class to each record, ultimately affecting the classification problem and metrics.

7.3 Limitation of this Research Work

The present study aimed to develop novel algorithms for mining data from heterogeneous distributed data streams. These algorithms were designed to acknowledge the saving of reserves for cases when there are limits on assets like communication and resource time. Most importantly, the resources were saved while not showing any detrimental effect on accuracy and gracefully maintaining the level of accuracy. Thereby, the research goals formulated at the commencement of this piece of work have been fruitfully addressed. Nevertheless, the achievement of this research work, some of the strategies used in this study, also enforce some restraints. A brief overview of these limitations has been listed below.

1. In PODSM, for generating the probabilistic view about the confidence status of another site, there should be sufficient statistics for each record schema by the trouble site. In fewer cases, even after obtaining the feedback data and applying the algorithm, the communication involved in the classification process stays

stable and is not much affected. However, this situation represents the nature of the data, which does not allow sufficient, efficient statistics to be learned by a trouble site, ultimately leading to the failure of a source site in estimating the probability of confidence status of another site. Moreover, in rare cases, even if it allows one to make computations and decisions accordingly, the decisions turn out to be erroneous, adversely affecting the accuracy of the classification procedure. One way to combat this limitation can be to modify the configuration parameters, like switching to other TF values to scrutinize if that change can allow the computation of efficient statistics, hence achieving a reduction in resources utilized while not producing much detrimental effect on accuracy.

2. The building block of the MTDM algorithm is the minimized tree, and its fundamental working lies in parsing this tree. In some cases, difficulty may arise in obtaining this tree. Due to the complexities of the dataset, a minimized centralization of records may not produce the minimized centralized tree. Hence, it would require a greater number of records to be centralized for generating a tree. Thus, the solution to learning the cross-terms between the heterogeneous distributed sites in this scenario comes with a considerable communication cost. The extreme case of this constraint would be when a central tree generation would require transmitting the complete set of records from all the primary sites. A quick fix for this concern can be to hyper-tune the parameters used for generating the tree. Nonetheless, this may or may not work, which would result in this algorithm being impotent in achieving efficiency in learning the cross-terms between distributed sites while requiring minimum additional communication.
3. For building a prediction model that can predict peak demand days for asthma-related hospitalizations, the list of predictors or input variables was enriched with novel attributes like Soil Deficit and UVI but still lacked the dominant factor

pollen. Pollen is one of the most common and critical triggers that can provoke asthma exacerbations. Many related studies have stated pollen as a chief indicator of an asthma attack. NZ does not have a monitoring system to detect pollen counts, thus preventing the illusion of this quantitative measure as a predictor in this study. In our work, the requirement was to obtain the daily quantitative reading for each parameter, which became unfeasible for pollen. Secondly, the ED visits and hospitalization data come only from one DHB within NZ and may not be generalizable to all NZ. Also, variation in the geographic area under study induces different correlation values with input parameters. Nevertheless, Auckland does have a similar deprivation profile to the rest of NZ and is the fourth largest DHB in NZ, thus providing a useful initial dataset for this work. The readings obtained for several parameters were from sites located at a distance from the Auckland Central region. In other cases, the sensor network measuring readings for parameters was dense, thus not providing the exact measurements according to the geographic segmentation. Consequently, a unified reading was used for each parameter for the Auckland region, although the study did not focus on adding the spatial location's impact on asthma.

7.4 Future Work

Considerably more work can be done to enhance the performance of mining knowledge from distributed data streams which would be a beneficial step towards saving resources in mining data from heterogeneous distributed data streams. We believe that the prediction model for asthma has the potential to serve as a helpful forecasting tool for health providers and organizations, so more modeling work can be conducted in this direction to improve the performance of this model. More broadly, future studies could devise and include some data preprocessing methodology at the primary sites for

PODSM and MTDM prior to using them in the classification process. Also, efforts can be made toward testing PODSM and MTDM on other real-world problems. A natural progression of this study can be undertaken in the future via the following potential ideas.

1. For PODSM, future work can be conducted by including confidence values with record feature value transmission to the trouble site. This would assist a trouble record in maintaining statistics, including confidence values which would eventually help in estimating the confidence value of a site for a record instead of approximating the probability of whether the other site would be confident or not. In addition, a message-passing technique can also be developed through which the sites may communicate their threshold confidence value to each other. This confidence value may be calculated using a recent window size that can be broadcasted in advance, allowing a site to establish a relationship between their local view and the communicated confidence value. Furthermore, in PODSM, the threshold of 0.5 was chosen as it represents a clear dividing line between confident and unconfident predictions. However, other threshold values can also be considered depending on the specific requirements and limitations of the system. In the current work, the algorithm was used for the classification problem, while it can also be adopted for other machine learning techniques like regression.
2. In the case of MTDM, an extra communication cost was added for generating a minimized tree to learn the cross-terms between the vertically distributed sites, which can serve as a fruitful area for future research. A greater focus on developing some mechanism to produce a central tree with less communication or even no communication of records could produce exciting findings. In other words, a methodology should be devised to help generate an approximation of a tree without communicating records. A path to achieve this can be generating

local trees at local sites using local data and communicating local trees to the centralized site for merging and approximating a centralized tree.

3. The work discussed in this thesis about the case study is the novel work done to explore population health prediction modeling for asthma, which has been a prominent health issue in NZ, providing an essential foundation for further investigation. This research has raised many questions for exploration in the future. Although several diverse novel parameters have been explored to study their correlation with asthma-related emergency hospitalizations, more social media data sources can be inspected to find their relationship with ED visits and asthma-related hospitalizations, such as Facebook, Twitter, and news blogs. This prediction model was developed with the focal point on non-communicable diseases as they do not require contact tracing. In this study, we emphasized on asthma disease, but in the future, this model can be tested on other non-communicable diseases, namely COPD, diabetes, and obesity. Also, further research can be carried out to test this model in clinical populations in a prospective manner. The findings of this research have triggered many challenges that may not be related to the indicated work, but more outstanding efforts can be undertaken towards them. As such, work can be done towards developing a model for obtaining measurements for daily pollen counts in NZ that would, in turn, be beneficial for predictive modeling targeting other respiratory diseases. Besides this, NZ has a dense sensor network for recording air quality parameters; interpolation-based algorithms can be designed to provide more readings by stratifying the geographic area, allowing for analyzing asthma-related hospitalizations according to well-defined spatial groups.

To put it in a nutshell, in this research work, we aimed to devise algorithms for mining heterogeneous distributed data streams, specifically for scenarios with resource

restrictions. The competency of these resource-efficient and cost-saver algorithms was tested on a variety of benchmark datasets. The results achieved on them and a real case study (for asthma predictive modeling) demonstrated the proficiency of algorithms to save resources while maintaining the accuracy of the mining process. We worked on a healthcare application area to validate these algorithms; however, there can be numerous domains where these algorithms can be applied to achieve fruitful results.

References

- Abdallah, Z. S., Gaber, M. M., Srinivasan, B. & Krishnaswamy, S. (2015, Feb). Adaptive mobile activity recognition system with evolving data streams. *Neurocomput.*, 150, 304–317. <https://doi.org/10.1016/j.neucom.2014.09.074>
- Abdel Wahab, O. (2022). Intrusion detection in the iot under data and concept drifts: Online deep learning approach. *IEEE Internet of Things Journal*, 9(20), 19706–19716. <https://doi.org/10.1109/JIOT.2022.3167005>
- Accordini, S., Corsico, A., Braggion, M., Gerbase, M., Gíslason, D., Gulsvik, A., ... de Marco, R. (2012, 09). The cost of persistent asthma in europe: An international population-based study in adults. *International archives of allergy and immunology*, 160, 93–101. <https://doi.org/10.1159/000338998>
- Adi, E., Anwar, A., Baig, Z. & Zeadally, S. (2020). Machine learning and data analytics for the iot. *Neural Computing and Applications*, 32(20), 16205–16233. <https://doi.org/10.1007/s00521-020-04874-y>
- Agarwal, S. (2013). Data mining: Data mining concepts and techniques. In *2013 international conference on machine intelligence and research advancement* (p. 203–207). <https://doi.org/10.1109/ICMIRA.2013.45>
- Aggarwal, C. (2007). *Data streams: Models and algorithms* (Vol. 31). <https://doi.org/10.1007/978-0-387-47534-9>
- Aggarwal, C. C. (2007). An introduction to data streams. In C. C. Aggarwal (Ed.), *Data streams: Models and algorithms* (pp. 1–8). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-47534-9_1
- Aggarwal, C. C., Han, J., Wang, J. & Yu, P. S. (2004). On demand classification of data streams. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining* (p. 503–508). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1014052.1014110>
- Aggarwal, C. C., Yu, P. S., Han, J. & Wang, J. (2003). A framework for clustering evolving data streams. In J.-C. Freytag, P. Lockemann, S. Abiteboul, M. Carey, P. Selinger & A. Heuer (Eds.), *Proceedings 2003 vldb conference* (p. 81–92). San Francisco: Morgan Kaufmann. <https://doi.org/10.1016/B978-012722442-8/50016-1>
- Agrawal, R., Imieliński, T. & Swami, A. (1993, jun). Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2), 207–216. <https://doi.org/10.1145/170036.170072>
- Akbar, A., Kousiouris, G., Pervaiz, H., Sancho, J., Ta-Shma, P., Carrez, F. & Moessner,

- K. (2018). Real-time probabilistic data fusion for large-scale iot applications. *IEEE Access*, 6, 10015-10027. <https://doi.org/10.1109/ACCESS.2018.2804623>
- Alaerjan, A. (2023). Towards sustainable distributed sensor networks: An approach for addressing power limitation issues in wsns. *Sensors*, 23(2). Retrieved from <https://www.mdpi.com/1424-8220/23/2/975> <https://doi.org/10.3390/s23020975>
- Alencar, B. M., Canário, J. P., Lobão Neto, R., Prazeres, C., Bifet, A. & Rios, R. A. (2023). Fog-deepstream: A new approach combining lstm and concept drift for data stream analytics on fog computing. *Internet of Things*, 22, 100731. Retrieved from <https://www.sciencedirect.com/science/article/pii/S2542660523000549> <https://doi.org/10.1016/j.iot.2023.100731>
- Alex, S. A. & Nayahi, J. J. V. (2020). Deep incremental learning for big data stream analytics. In A. Pandian, T. Senjyu, S. M. S. Islam & H. Wang (Eds.), *Proceeding of the international conference on computer networks, big data and iot (iccbi - 2018)* (pp. 600–614). Cham: Springer International Publishing.
- Alhaj Ali, A., Vařacha, P., Urbanek, A., Krayem, S. & Žáček, P. (2018). Distributed data mining systems: techniques, approaches and algorithms. *MATEC Web of Conferences*, 210. <https://doi.org/10.1051/mateconf/201821004038>
- Al-Kateb, M. & Lee, B. S. (2014). Adaptive stratified reservoir sampling over heterogeneous data streams. *Information Systems*, 39, 199-216. <https://doi.org/10.1016/j.is.2012.03.005>
- Araz, O., Bentley, D. & Muelleman, R. (2014). Using google flu trends data in forecasting influenza-like-illness related emergency department visits in omaha, nebraska. *The American Journal of Emergency Medicine*, 32. <http://doi.org/10.1016/j.ajem.2014.05.052>
- Aronis, J., Kolluri, V., Provost, F. & Buchanan, B. (1997). The world: Knowledge discovery from multiple distributed databases. In *Proceedings of 10th international florida ai research symposium* (pp. 337–341).
- Arya, M. & G, H. S. (2022). Ensemble federated learning for classifying iomt data streams. In *2022 ieee 7th international conference for convergence in technology (i2ct)* (p. 1-5). <https://doi.org/10.1109/I2CT54291.2022.9824145>
- Asheim, A., Bjørnsen, L., Næss-Pleym, L., Uleberg, O., Dale, J. & Nilsen, S. (2019). Real-time forecasting of emergency department arrivals using prehospital data. *BMC Emergency Medicine*, 19. <http://doi.org/10.1186/s12873-019-0256-z>
- Asthma & of America, A. F. (2017). *Asthma*. Retrieved 12.12.2020, from <https://www.aafa.org/weather-triggers-asthma/>
- Asthma Respiratory Foundation NZ. (2019). *Asthma Infographic Impact Report 2019*. Retrieved 11-11-2020, from <https://www.asthmafoundation.org.nz/resources/asthma-in-new-zealand>
- Asthma Respiratory Foundation NZ. (2021). *Respiratory disease in new zealand*. Retrieved from <https://www.asthmafoundation.org.nz/research/key-statistics>
- Babin, S., Burkom, H., Holtry, R., Tabernero, N., Stokes, L., Davies-Cole, J., ... Lee, D. (2007). Pediatric patient asthma-related emergency department visits and

- admissions in washington, dc, from 2001–2004, and associations with air quality, socio-economic status and age group. *Environmental health : a global access science source*, 6, 9. <http://doi.org/10.1186/1476-069X-6-9>
- Bandyopadhyay, S., Giannella, C., Maulik, U., Kargupta, H., Liu, K. & Datta, S. (2006). Clustering distributed data streams in peer-to-peer environments. *Inf. Sci.*, 176(14), 1952–1985. <https://doi.org/10.1016/j.ins.2005.11.007>
- Bannister, T., Ebert, E. E., Williams, T., Douglas, P., Wain, A., Carroll, M., ... others (2021). A pilot forecasting system for epidemic thunderstorm asthma in southeastern australia. *Bulletin of the American Meteorological Society*, 102(2), E399–E420. <https://doi.org/10.1175/BAMS-D-19-0140.1>
- Barnard, L. T. & Zhang, J. (2019). *The impact of respiratory disease in new zealand: 2018 update*. Asthma and Respiratory Foundation NZ.
- Barnard, L. T. & Zhang, J. (2021). *The impact of respiratory disease in new zealand: 2020 update*. University of Otago: Dunedin, New Zealand.
- Basak, J. & Kothari, R. (2004, 08). A classification paradigm for distributed vertically partitioned data. *Neural computation*, 16, 1525–44. <https://doi.org/10.1162/089976604323057470>
- Bascom, R., Bromberg, P., Costa, D., Devlin, R., Dockery, D., Frampton, M., ... Utell, M. (1996). Health effects of outdoor air pollution. *American Review of Respiratory Disease*, 153(2), 477–498. <https://doi.org/10.1164/ajrccm.153.2.8564086>
- Becker, S. & Soukup, J. M. (1999). Exposure to urban air particulates alters the macrophage-mediated inflammatory response to respiratory viral infection. *Journal of Toxicology and Environmental Health, Part A*, 57(7), 445–457. <https://doi.org/10.1080/009841099157539>
- Beretta, S., Castelli, M., Gonçalves, I. & Ramazzotti, D. (2017). A quantitative assessment of the effect of different algorithmic schemes to the task of learning the structure of bayesian networks. *CoRR*, abs/1704.08676. <https://doi.org/10.48550/arXiv.1704.08676>
- Beringer, J. & Hüllermeier, E. (2007). An efficient algorithm for instance-based learning on data streams. In *Proceedings of the 7th industrial conference on advances in data mining: Theoretical aspects and applications* (Vol. 4597, p. 34–48). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-73435-2_4
- Bertoni, A. & Larsson, T. (2017). Data mining in product service systems design: Literature review and research questions. *Procedia CIRP*, 64, 306–311. Retrieved from <https://www.sciencedirect.com/science/article/pii/S2212827117302809> (9th CIRP IPSS Conference: Circular Perspectives on PSS) <https://doi.org/10.1016/j.procir.2017.03.131>
- Bessa, R. J., Miranda, V. & Gama, J. (2009). Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting. *IEEE Transactions on Power Systems*, 24(4), 1657–1666. <https://doi.org/10.1109/TPWRS.2009.2030291>
- Beyer, J., Heesche, K., Hauptmann, W., Otte, C. & Kruse, R. (2009). Ensemble learning for multi-source information fusion. In *Proceedings of the 10th european conference on symbolic and quantitative approaches to reasoning with uncertainty*

- (p. 748–756). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-02906-6_64
- Bhaduri, K., Das, K., Sivakumar, K., Kargupta, H., Wolff, R. & Chen, R. (2007). Algorithms for distributed data stream mining. In C. C. Aggarwal (Ed.), *Data streams: Models and algorithms* (pp. 309–331). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-47534-9_14
- Bhaduri, K. & Srivastava, A. N. (2009). A local scalable distributed expectation maximization algorithm for large peer-to-peer networks. In *2009 ninth ieee international conference on data mining* (p. 31–40). <https://doi.org/10.1109/ICDM.2009.45>
- Bhavsar, N., Gao, A., Phelan, M., Pagidipati, N. & Goldstein, B. (2018). Value of neighborhood socioeconomic status in predicting risk of outcomes in studies that use electronic health record data. *JAMA Network Open*, 1(5). <http://doi.org/10.1001/jamanetworkopen.2018.2716>
- Bifet, A. & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In (Vol. 7, pp. 443–448). <https://doi.org/10.1137/1.9781611972771.42>
- Bifet, A., Gavaldà, R., Holmes, G. & Pfahringer, B. (2018). Classification. In *Machine Learning for Data Streams: with Practical Examples in MOA*. The MIT Press. <https://doi.org/10.7551/mitpress/10654.003.0011>
- Bifet, A., Holmes, G. & Pfahringer, B. (2010). Leveraging bagging for evolving data streams. In *Proceedings of the 2010th european conference on machine learning and knowledge discovery in databases - volume part I* (p. 135–150). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-15880-3_15
- Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R. & Gavaldà, R. (2009). New ensemble methods for evolving data streams. In *Proceedings of the 15th acm sigkdd international conference on knowledge discovery and data mining* (p. 139–148). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1557019.1557041>
- Bifet, A. & Kirkby, R. (2009). Data stream mining a practical approach..
- Bifet, A., Read, J., Žliobaitė, I., Pfahringer, B. & Holmes, G. (2013). Pitfalls in benchmarking data stream classification and how to avoid them. In H. Blockeel, K. Kersting, S. Nijssen & F. Železný (Eds.), *Machine learning and knowledge discovery in databases* (pp. 465–479). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-40988-2_30
- Blakey, J. D., Price, D. B., Pizzichini, E., Popov, T. A., Dimitrov, B. D., Postma, D. S., ... Thomas, M. (2017). Identifying risk of future asthma attacks using uk medical record data: A respiratory effectiveness group initiative. *The Journal of Allergy and Clinical Immunology: In Practice*, 5(4), 1015–1024.e8. <https://doi.org/10.1016/j.jaip.2016.11.007>
- Borgi, T., Zoghلامي, N., Abed, M. & Mohamed Saber, N. (2017, 07). Big data for operational efficiency of transport and logistics: A review. In (p. 113–120). <https://doi.org/10.1109/ICAdLT.2017.8547029>
- Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Solano, F. E. & Rendon, O. (2018). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services*

- and Applications*, 9(1), 1-99. <https://doi.org/10.1186/s13174-018-0087-2>
- Bradley, S. (2022). *How common is asthma? worldwide facts and statistics*. Retrieved 09-04-2022, from <https://www.singlecare.com/blog/asthma-statistics/>
- Brameld, K. J. & Holman, C. D. J. (2006). Demographic factors as predictors for hospital admission in patients with chronic disease. *Australian and New Zealand Journal of Public Health*, 30(6), 562-566. <https://doi.org/10.1111/j.1467-842X.2006.tb00787.x>
- Bridge, J., Blakey, J. D. & Bonnett, L. J. (2020). A systematic review of methodology used in the development of prediction models for future asthma exacerbation. *BMC Medical Research Methodology*, 20(1). <https://doi.org/10.1186/s12874-020-0913-7>
- Brooks, R., Ramanathan, P. & Sayeed, A. (2003). Distributed target classification and tracking in sensor networks. *Proceedings of the IEEE*, 91(8), 1163-1171. <https://doi.org/10.1109/JPROC.2003.814923>
- Buntine, W. L. (2013). Theory refinement on bayesian networks. *CoRR, abs/1303.5709*. <https://doi.org/10.48550/arXiv.1303.5709>
- Burnett, R. T., Cakmak, S., Brook, J. R. & Krewski, D. (1997). The role of particulate size and chemistry in the association between summertime ambient air pollution and hospitalization for cardiorespiratory diseases. *Environmental Health Perspectives*, 105(6), 614 - 620. <https://doi.org/10.1289/ehp.97105614>
- Buyuktiryaki, B., Civelek, E., Can, D., Orhan, F., Aydogan, M., Reisli, I., ... Sekerel, B. (2013). Predicting hospitalization in children with acute asthma. *The Journal of emergency medicine*, 44(5), 919-927. <https://doi.org/10.1016/j.jemermed.2012.10.015>
- Byers, N., Ritchey, M., Vaidyanathan, A., Brandt, A. & Yip, F. (2015). Short-term effects of ambient air pollutants on asthma-related emergency department visits in indianapolis, indiana, 2007-2011. *The Journal of asthma : official journal of the Association for the Care of Asthma*, 53(3), 1-8. <https://doi.org/10.3109/02770903.2015.1091006>
- Bühlmann, P. (2012). Bagging, boosting and ensemble methods. In J. E. Gentle, W. K. Härdle & Y. Mori (Eds.), *Handbook of computational statistics: Concepts and methods* (pp. 985-1022). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-21551-3_33
- Cano, A. & Krawczyk, B. (2020). Kappa updated ensemble for drifting data stream mining. *Mach. Learn.*, 109(1), 175-218. <https://doi.org/10.1007/s10994-019-05840-z>
- Cantoni, V., Lombardi, L. & Lombardi, P. (2006). Challenges for data mining in distributed sensor networks. In *18th international conference on pattern recognition (icpr'06)* (Vol. 1, pp. 1000-1007). <https://doi.org/10.1109/ICPR.2006.359>
- Canzian, L., Zhang, Y. & van der Schaar, M. (2013). Ensemble of distributed learners for online classification of dynamic data streams. *CoRR, abs/1308.5281*. <https://doi.org/10.48550/arXiv.1308.5281>

- Carlos-Mancilla, M., López-Mellado, E. & Siller, M. (2016). Wireless sensor networks formation: approaches and techniques. *Journal of Sensors*, 2016, 1-18. <https://doi.org/10.1155/2016/2081902>
- Chan, P., Fan, W., Prodrumidis, A. & Stolfo, S. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and their Applications*, 14(6), 67-74. <https://doi.org/10.1109/5254.809570>
- Chandola, V., Banerjee, A. & Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3). <https://doi.org/10.1145/1541880.1541882>
- Chapelle, O. & Chang, Y. (2011). Yahoo! learning to rank challenge overview. In O. Chapelle, Y. Chang & T.-Y. Liu (Eds.), *Proceedings of the learning to rank challenge* (Vol. 14, pp. 1-24). PMLR.
- Charlson, M. E., Pompei, P., Ales, K. L. & MacKenzie, C. (1987). A new method of classifying prognostic comorbidity in longitudinal studies: Development and validation. *Journal of Chronic Diseases*, 40(5), 373-383. [https://doi.org/10.1016/0021-9681\(87\)90171-8](https://doi.org/10.1016/0021-9681(87)90171-8)
- Chauhan, A., Krishna, M., Frew, A. & Holgate, S. (1998). Exposure to nitrogen dioxide (no2) and respiratory disease risk. *Reviews on environmental health*, 13(1-2), 73-90.
- Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, W. (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16, 321-357. <https://doi.org/10.1613/jair.953>
- Chawla, N. V., Hall, L. O., Bowyer, K. W. & Kegelmeyer, W. P. (2004). Learning ensembles from bites: A scalable and accurate approach. *J. Mach. Learn. Res.*, 5, 421-451.
- Chen, C., Weider, K., Konopka, K. & Danis, M. (2014). Incorporation of socioeconomic status indicators into policies for the meaningful use of electronic health records. *Journal of health care for the poor and underserved*, 25(1), 1-16. <https://doi.org/10.1353/hpu.2014.0040>
- Chen, L. D., Sakaguchi, T. & Frolick, M. N. (2000). Data mining methods, applications, and tools. *IS Management*, 17(1), 1-6. <https://doi.org/10.1201/1078/43190.17.1.20000101/31216.9>
- Chen, R., Sivakumar, K. & Kargupta, H. (2001). An approach to online bayesian learning from multiple data streams. In *Proceedings of workshop on mobile and distributed data mining, pkdd* (Vol. 1, pp. 31-45).
- Chen, R., Sivakumar, K. & Kargupta, H. (2004). Collective mining of bayesian networks from distributed heterogeneous data. *Knowledge and Information Systems*, 6(2), 164-187. <https://doi.org/10.1007/BF02637154>
- Chen, S. & Wu, S. (2019). Deep learning for identifying environmental risk factors of acute respiratory diseases in beijing, china: implications for population with different age and gender. *International Journal of Environmental Health Research*, 30(4), 435-446. <https://doi.org/10.1080/09603123.2019.1597836>
- Chen, X. & Rowe, N. C. (2011). An energy-efficient communication scheme in wireless cable sensor networks. In *2011 IEEE International Conference on Communications (ICC)* (p. 1-5). <https://doi.org/10.1109/icc.2011.5963077>

- Chmaj, G. & Selvaraj, H. (2015). Distributed processing applications for uav/drones: A survey. In H. Selvaraj, D. Zydek & G. Chmaj (Eds.), *Progress in systems engineering* (pp. 449–454). Cham: Springer International Publishing.
- Cho, K., Ilin, A. & Raiko, T. (2011). Improved learning of gaussian-bernoulli restricted boltzmann machines. In *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I* (p. 10–17). Berlin, Heidelberg: Springer.
- Chung, H. M. & Gray, P. (1999). Special section: Data mining. *Journal of Management Information Systems*, 16(1), 11–16. <https://doi.org/10.1080/07421222.1999.11518231>
- Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X. & Zhu, M. (2002). Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations Newsletter*, 4(2), 28–34. <https://doi.org/10.1145/772862.772867>
- Cochran, W. G. (1977). *Sampling techniques*, 3rd edition. John Wiley.
- Cormode, G., Muthukrishnan, S. & Zhuang, W. (2007). Conquering the divide: Continuous clustering of distributed data streams. In *2007 IEEE 23rd International Conference on Data Engineering* (p. 1036–1045). <https://doi.org/10.1109/ICDE.2007.368962>
- Custovic, A., Johnston, S., Pavord, I., Gaga, M., Fabbri, L., Bel, E., ... Papadopoulos, N. (2013, December). Eaaci position statement on asthma exacerbations and severe asthma. *Allergy*, 68(12), 1520–1531. <https://doi.org/10.1111/all.12275>
- Daines, L., Mclean, S., Buelo, A., Lewis, S., Sheikh, A. & Pinnock, H. (2018). Clinical prediction models to support the diagnosis of asthma in primary care: A systematic review protocol. *npj Primary Care Respiratory Medicine*, 28(1), 1–4. <https://doi.org/10.1038/s41533-018-0086-6>
- da Silva, J. C., Giannella, C., Bhargava, R., Kargupta, H. & Klusch, M. (2005). Distributed data mining and agents. *Eng. Appl. Artif. Intell.*, 18(7), 791–807. <https://doi.org/10.1016/j.engappai.2005.06.004>
- Datar, M., Gionis, A., Indyk, P. & Motwani, R. (2002). Maintaining stream statistics over sliding windows: (extended abstract). In *Proceedings of the Thirteenth Annual Acm-Siam Symposium on Discrete Algorithms* (p. 635–644). USA: Society for Industrial and Applied Mathematics. <https://doi.org/10.1145/545381.545466>
- Datta, S., Bhaduri, K., Giannella, C., Wolff, R. & Kargupta, H. (2006). Distributed data mining in peer-to-peer networks. *IEEE Internet Computing*, 10(4), 18–26. <https://doi.org/10.1109/MIC.2006.74>
- Dattagupta, S. J. (2018). *A performance comparison of oversampling methods for data generation in imbalanced learning tasks* (Unpublished master's thesis). Lisboa PT.
- De Francisci Morales, G., Bifet, A., Khan, L., Gama, J. & Fan, W. (2016). IoT big data stream mining. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 2119–2120). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2939672.2945385>
- Denham, B. (2019). *Novel methods for distributed and privacy-preserving data stream*

- mining* (Unpublished master's thesis). Auckland University of Technology, Auckland, NZ.
- Denham, B., Pears, R. & Naeem, M. A. (2020a). Enhancing random projection with independent and cumulative additive noise for privacy-preserving data stream mining. *Expert Systems with Applications*, 152, 113380. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0957417420302049> <https://doi.org/10.1016/j.eswa.2020.113380>
- Denham, B., Pears, R. & Naeem, M. A. (2020b). HDSM: A distributed data mining approach to classifying vertically distributed data streams. *Knowledge-Based Systems*, 189. <https://doi.org/10.1016/j.knosys.2019.105114>
- Desikan, P. K., Srivastava, N., Winden, T., Lindquist, T., Britt, H. & Srivastava, J. (2012). Early prediction of potentially preventable events in ambulatory care sensitive admissions from clinical data. In *2012 IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology* (p. 124). IEEE Computer Society. <https://doi.org/10.1109/HISB.2012.49>
- Devi, S. (2014). A survey on distributed data mining and its trends. *International Journal of Research in Engineering & Technology (IMPACT: IJRET)*, 2(3), 107–120.
- Ditzler, G. & Polikar, R. (2013). Incremental learning of concept drift from streaming imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 25(10), 2283-2301. <https://doi.org/10.1109/TKDE.2012.136>
- Divo, M. J., Martinez, C. H. & Mannino, D. M. (2014). Ageing and the epidemiology of multimorbidity. *European Respiratory Journal*, 44(4), 1055 - 1068. <https://doi.org/10.1183/09031936.00059814>
- Do, Q., Doig, A. & Son, T. (2019). Deep Q-learning for predicting asthma attack with considering personalized environmental triggers' risk scores. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Vol. 2019, p. 562-565). <https://doi.org/10.1109/EMBC.2019.8857172>
- Dobra, A., Garofalakis, M., Gehrke, J. & Rastogi, R. (2002). Processing complex aggregate queries over data streams. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data* (p. 61–72). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/564691.564699>
- Dobra, A., Garofalakis, M., Gehrke, J. & Rastogi, R. (2004). Sketch-based multi-query processing over data streams. In *Advances in Database Technology - EDBT 2004* (Vol. 2992, p. 551-568). https://doi.org/10.1007/978-3-540-24741-8_32
- Domingos, P. & Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 71–80). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/347090.347107>
- Dong, J., Zhuang, D., Huang, Y. & Fu, J. (2009). Advances in multi-sensor data fusion: Algorithms and applications. *Sensors*, 9(10), 7771 - 7784. <https://doi.org/10.3390/s91007771>
- Dua, D. & Graff, C. (2017). *UCI machine learning repository*. Retrieved from

- <http://archive.ics.uci.edu/ml>
- Ducret-Stich, R. E., Tsai, M.-Y., Ragetti, M. S., Ineichen, A., Kuenzli, N. & Phuleria, H. C. (2013). Role of highway traffic on spatial and temporal distributions of air pollutants in a swiss alpine valley. *Science of The Total Environment*, 456-457, 50-60. <https://doi.org/10.1016/j.scitotenv.2013.03.065>
- Duda, R., Hart, P. & Stork, D. (2012). *Pattern classification*. Wiley. Retrieved from <https://books.google.co.nz/books?id=Br33IRC3PkQC>
- Elliot, A., Bone, A., Morbey, R., Hughes, H., Harcourt, S., Smith, S., ... Smith, G. (2014). Using real-time syndromic surveillance to assess the health impact of the 2013 heatwave in england. *Environmental Research*, 135, 31–36. <https://doi.org/10.1016/j.envres.2014.08.031>
- Elliot, A. J., Hughes, H. E., Hughes, T. C., Locker, T. E., Brown, R., Saran, C., ... Smith, G. E. (2014). The impact of thunderstorm asthma on emergency department attendances across london during july 2013. *Emergency Medicine Journal*, 31(8), 675–678. <https://doi.org/10.1136/emmermed-2013-203122>
- Enilari, O. & Sinha, S. (2019, 01). The global impact of asthma in adult populations. *Annals of Global Health*, 85. <https://doi.org/10.5334/aogh.2412>
- Fan, B., Peng, J., Guo, H., Gu, H., Xu, K. & Wu, T. (2022). Accurate forecasting of emergency department arrivals with internet search index and machine learning models: Model development and performance evaluation. *JMIR Med Inform*, 10(7). <https://doi.org/10.2196/34504>
- Fayaz, M., Khan, A., Rahman, J., Uddin, M. I. & Alouffi, B. (2020). Ensemble machine learning model for classification of spam product reviews. *Complexity*, 1-10. <https://doi.org/10.1155/2020/8857570>
- Fernández-Delgado, M., Cernadas, E., Barro, S. & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15(1), 3133-3181. Retrieved from <http://jmlr.org/papers/v15/delgado14a.html> <https://doi.org/10.1117/1.JRS.11.015020>
- Forno, E., Fuhlbrigge, A., Soto-Quiros, M., Avila, L., Raby, B., Brehm, J., ... Celedon, J. (2010). Risk factors and predictive clinical scores for asthma exacerbations in childhood. *Chest*, 138, 1156-65. <https://doi.org/10.1378/chest.09-2426>
- Friedman, N. & Goldszmidt, M. (1997). Sequential update of bayesian network structure. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence* (p. 165–174). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Fu, Y. (2001). Distributed data mining: An overview. *Newsletter of the IEEE Technical Committee on Distributed Processing*, 5(9).
- Fuqiang, Y. (2011). The research on distributed data stream mining based on mobile agent. *Procedia Engineering*, 23, 103-108. <https://doi.org/10.1016/j.proeng.2011.11.2473>
- Fusco, D., Forastiere, F., Michelozzi, P., Spadea, T., Ostro, B., Arcà, M. & Perucci, C. (2001). Air pollution and hospital admissions for respiratory conditions

- in Rome, Italy. *The European Respiratory Journal*, 17, 1143-1150. <https://doi.org/10.1183/09031936.01.00005501>
- Gaber, M., Stahl, F. & Gomes, J. (2014). Pocket data mining framework. In (Vol. 2, p. 23-40). https://doi.org/10.1007/978-3-319-02711-1_3
- Gaber, M. M., Krishnaswamy, S. & Zaslavsky, A. B. (2004). A wireless data stream mining model. In Q. H. Mahmoud & H. Weghorn (Eds.), *Wireless information systems, Proceedings of the 3rd International Workshop on Wireless Information Systems, WIS 2004, In conjunction with ICEIS 2004, Porto, Portugal, April 2004* (pp. 152–160). INSTICC Press.
- Gaber, M. M., Zaslavsky, A. & Krishnaswamy, S. (2005). Mining data streams: A review. *SIGMOD Rec.*, 34(2), 18–26. <https://doi.org/10.1145/1083784.1083789>
- Gama, J. a. & Kosina, P. (2014). Recurrent concepts in data streams classification. *Knowl. Inf. Syst.*, 40(3), 489–507. <https://doi.org/10.1007/s10115-013-0654-6>
- Gama, J. a., Rocha, R. & Medas, P. (2003). Accurate decision trees for mining high-speed data streams. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 523–528). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/956750.956813>
- Gama, J. a., Žliobaitundefined, I., Bifet, A., Pechenizkiy, M. & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4). <https://doi.org/10.1145/2523813>
- Gan, W., Lin, C.-W., Chao, H.-C. & Zhan, J. (2017). Data mining in distributed environment: a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(6). <https://doi.org/10.1002/widm.1216>
- Ganaie, M., Hu, M., Malik, A., Tanveer, M. & Suganthan, P. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115. <https://doi.org/10.1016/j.engappai.2022.105151>
- Garg, M. K., Kim, D.-J., Turaga, D. S. & Prabhakaran, B. (2010). Multimodal analysis of body sensor network data streams for real-time healthcare. In *Proceedings of the International Conference on Multimedia Information Retrieval* (p. 469–478). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1743384.1743467>
- Garofalakis, M. (2009). Distributed data streams. In L. LIU & M. T. ÖZSU (Eds.), *Encyclopedia of Database Systems* (pp. 883–890). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-39940-9_137
- Geng, L. & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3). <https://doi.org/10.1145/1132960.1132963>
- Gibbons, P. B. & Matias, Y. (1998). New sampling-based summary statistics for improving approximate query answers. *SIGMOD Rec.*, 27(2), 331–342. <https://doi.org/10.1145/276305.276334>
- Giraud-Carrier, C. (2000). A note on the utility of incremental learning. *AI Communications*, 13(4), 215–223.
- The Global Asthma Report 2018*. (2018). Global Asthma Network. Retrieved from <http://globalasthmareport.org/resources/>

- Global_Asthma_Report_2018.pdf ([Online; accessed 19-October-2021])
- Gomes, H. M., Barddal, J. P., Enembreck, F. & Bifet, A. (2017). A survey on ensemble learning for data stream classification. *ACM Comput. Surv.*, 50(2). <https://doi.org/10.1145/3054925>
- Gomes, H. M. & Enembreck, F. (2014). SAE2: Advances on the social adaptive ensemble classifier for data streams. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing* (p. 798–804). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2554850.2554905>
- Gomes, M. M., da Rosa Righi, R., André da Costa, C. & Griebler, D. (2021). Simplifying iot data stream enrichment and analytics in the edge. *Computers & Electrical Engineering*, 92, 107110. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0045790621001154> <https://doi.org/10.1016/j.compeleceng.2021.107110>
- Gorawski, M. & Płuciennik-Psota, E. (2009). Distributed data mining methodology with classification model example. In N. T. Nguyen, R. Kowalczyk & S.-M. Chen (Eds.), *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems* (pp. 107–117). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Gunawardana, A., Meek, C. & Xu, P. (2011). A model for temporal dependencies in event streams. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 24). Curran Associates, Inc.
- Habib ur Rehman, M., Khan, A. u. R. & Batool, A. (2016). Big data analytics in mobile and cloud computing environments. In *Innovative research and applications in next-generation high performance computing* (p. 359-377). <https://doi.org/10.4018/978-1-5225-0287-6.ch014>
- Habib ur Rehman, M., Liew, C. S., Wah, T., Shuja, J. & Daghighi, B. (2015). Mining personal data using smartphones and wearable devices: A survey. *Sensors*, 15, 4430-4469. <https://doi.org/10.3390/s150204430>
- Hallit, S., Raherison, C., Malaeb, D., Hallit, R., Waked, M., Kheir, N. & Salameh, P. (2019). Development of an asthma risk factors scale (arfs) for risk assessment asthma screening in children. *Pediatrics & Neonatology*, 60(2), 156-165. <https://doi.org/10.1016/j.pedneo.2018.05.009>
- Halonen, J. I., Lanki, T., Yli-Tuomi, T., Kulmala, M., Tiittanen, P. & Pekkanen, J. (2008). Urban air pollution, and asthma and copd hospital emergency room visits. *Thorax*, 63(7), 635 - 641. <https://doi.org/10.1136/thx.2007.091371>
- Halterman, J., Borrelli, B., Tremblay, P., Conn, K., Fagnano, M., Montes, G. & Hernandez, T. (2009). Screening for environmental tobacco smoke exposure among inner-city children with asthma. *Pediatrics*, 122, 1277-1283. <https://doi.org/10.1542/peds.2008-0104>
- Han, J., Pei, J. & Yin, Y. (2000). Mining frequent patterns without candidate generation. *SIGMOD Rec.*, 29(2), 1–12. <https://doi.org/10.1145/335191.335372>
- Hand, D. J., Smyth, P. & Mannila, H. (2001). *Principles of data mining*. Cambridge,

- MA, USA: MIT Press.
- Hanson, M. A., Powell Jr., H. C., Barth, A. T., Ringgenberg, K., Calhoun, B. H., Aylor, J. H. & Lach, J. (2009). Body area sensor networks: Challenges and opportunities. *Computer*, 42(1), 58-65. <https://doi.org/10.1109/MC.2009.5>
- Haque, A., Khan, L. & Baron, M. (2016). SAND: Semi-supervised adaptive novel class detection and classification over data stream. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (p. 1652–1658). AAAI Press.
- Harper, A. & Mustafee, N. (2019). A hybrid modelling approach using forecasting and real-time simulation to prevent emergency department overcrowding. In *2019 Winter Simulation Conference (WSC)* (p. 1208-1219). <https://doi.org/10.1109/WSC40007.2019.9004862>
- Harrou, F., Dairi, A., Kadri, F. & Sun, Y. (2020). Forecasting emergency department overcrowding: A deep learning framework. *Chaos, Solitons & Fractals*, 139. <https://doi.org/10.1016/j.chaos.2020.110247>
- Harvey, J. L. & Kumar, S. A. P. (2019). Machine learning for predicting development of asthma in children. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)* (p. 596-603). <https://doi.org/10.1109/SSCI44817.2019.9002692>
- Hassani, A., Montori, F., Liao, K., Haghighi, P. D., Jayaraman, P. P., Georgakopoulos, D., ... Zaslavsky, A. (2021). Inform: A tool for classification and semantic annotation of iot datastreams. In *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)* (p. 223-228). <https://doi.org/10.1109/WF-IoT51360.2021.9594994>
- Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301), 13-30. <https://doi.org/10.1080/01621459.1963.10500830>
- Homayoun, S. & Ahmadzadeh, M. (2016). A review on data stream classification approaches. *Journal of Advanced Computer Science & Technology*, 5, 8. <https://doi.org/10.14419/jacst.v5i1.5225>
- Hoot, N. R., Zhou, C., Jones, I. & Aronsky, D. (2007). Measuring and forecasting emergency department crowding in real time. *Annals of emergency medicine*, 49(6), 747–755. <https://doi.org/10.1016/j.annemergmed.2007.01.017>
- Hsu, C. & Wallace, W. (2007). An industrial network flow information integration model for supply chain management and intelligent transportation. *Enterprise Information Systems*, 1. <https://doi.org/10.1080/17517570701504633>
- Hui, C. Y., Walton, R., McKinstry, B., Jackson, T., Parker, R. & Pinnock, H. (2016). The use of mobile applications to support self-management for people with asthma: a systematic review of controlled studies to identify features associated with clinical effectiveness and adherence. *Journal of the American Medical Informatics Association*, 24(3), 619-632. <https://doi.org/10.1093/jamia/ocw143>
- Hulten, G., Spencer, L. & Domingos, P. (2001). Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 97–106). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/502512.502529>
- IBM Cloud Education. (2021). *Data mining*. Retrieved 2021-12-05, from <https://www.ibm.com/cloud/learn/data-mining>

- Idrees, M. M., Minku, L. L., Stahl, F. & Badii, A. (2020). A heterogeneous online learning ensemble for non-stationary environments. *Knowledge-Based Systems*, 188, 104983. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0950705119304058> <https://doi.org/10.1016/j.knosys.2019.104983>
- Ikonomovska, E., Gama, J., Sebastião, R. & Gjorgjevik, D. (2009). Regression trees from data streams with drift detection. In J. Gama, V. S. Costa, A. M. Jorge & P. B. Brazdil (Eds.), *Discovery science* (pp. 121–135). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Ikonomovska, E., Loskovska, S. & Gjorgjevikj, D. (2007). A survey of stream data mining. In *Proceedings of 8th National Conference with International Participation, ETAI* (p. 19-21).
- Ilin, I. & Klimin, A. (2019, 01). Features of big data approach and new opportunities of bi-systems in marketing activities. *E3S Web of Conferences*, 110, 02054. <https://doi.org/10.1051/e3sconf/201911002054>
- Janardan & Mehta, S. (2017). Concept drift in streaming data classification: Algorithms, platforms and issues. *Procedia Computer Science*, 122, 804-811. <https://doi.org/10.1016/j.procs.2017.11.440>
- Jethva, H. (2022). *Distributed data mining: 7 critical processes & algorithms*. Retrieved from <https://hevodata.com/learn/distributed-data-mining/> (01/11/2022)
- Jia, S. (2020). A VFDT algorithm optimization and application thereof in data stream classification. *Journal of Physics: Conference Series*, 1629(1). <https://doi.org/10.1088/1742-6596/1629/1/012027>
- Jiao, Y. (2006). Maintaining stream statistics over multiscale sliding windows. *ACM Trans. Database Syst.*, 31(4), 1305–1334. <https://doi.org/10.1145/1189769.1189773>
- Joyce, J. (2021). Bayes' Theorem. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Fall 2021 ed.). Metaphysics Research Lab, Stanford University. Retrieved from <https://plato.stanford.edu/archives/fall2021/entries/bayes-theorem/>
- Kalra, M. & Lal, N. (2016). Data mining of heterogeneous data with research challenges. In *2016 Symposium on Colossal Data Analysis and Networking (CDAN)* (p. 1-6). <https://doi.org/10.1109/CDAN.2016.7570899>
- Kargupta, H., Bhargava, R., Liu, K., Powers, M., Blair, P., Bushra, S., ... Handy, D. (2004). VEDAS: A mobile and distributed data stream mining system for real-time vehicle monitoring. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)* (p. 300-311). <https://doi.org/10.1137/1.9781611972740.28>
- Kargupta, H., Byung-Hoon, Hershberger, D. & Johnson, E. (1999). Collective data mining: A new perspective toward distributed data analysis. In *Advances in Distributed and Parallel Knowledge Discovery* (pp. 133–184). Cambridge, MA, United States: MIT Press.
- Karnick, M., Ahiskali, M., Muhlbaier, M. D. & Polikar, R. (2008). Learning concept

- drift in nonstationary environments using an ensemble of classifiers based approach. In *2008 IEEE International Joint Conference on Neural Networks* (p. 3455-3462). <https://doi.org/10.1109/IJCNN.2008.4634290>
- Keahey, L., Bulloch, B., Becker, A. B., Pollack, C. V., Clark, S. & Camargo, C. A. (2002). Initial oxygen saturation as a predictor of admission in children presenting to the emergency department with acute asthma. *Annals of Emergency Medicine*, 40(3), 300-307. <https://doi.org/10.1067/mem.2002.126813>
- Kesavaraj, G. & Sukumaran, S. (2013). A study on classification techniques in data mining. In *2013 fourth international conference on computing, communications and networking technologies (icccnt)* (p. 1-7). 10.1109/ICCCNT.2013.6726842
- Khan, L. & Fan, W. (2012). Tutorial: Data stream mining and its applications. In S.-g. Lee, Z. Peng, X. Zhou, Y.-S. Moon, R. Unland & J. Yoo (Eds.), *Database Systems for Advanced Applications* (pp. 328–329). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Khatri, K. & Tamil, L. (2017). Early detection of peak demand days of chronic respiratory diseases emergency department visits using artificial neural networks. *IEEE Journal of Biomedical and Health Informatics*, 22(1), 285-290. <https://doi.org/10.1109/JBHI.2017.2698418>
- Khedo, K. K., Perseedoss, R. & Mungur, A. (2010). A wireless sensor network air pollution monitoring system. *International Journal of Wireless & Mobile Networks*, 2(2), 31-45. <https://doi.org/10.5121/ijwmn.2010.2203>
- Kholod, I., Kuprianov, M., Titkov, E., Shorov, A., Postnikov, E., Mironenko, I. & Sokolov, S. (2019). Training normal bayes classifier on distributed data. *Procedia Computer Science*, 150, 389-396. <https://doi.org/10.1016/j.procs.2019.02.068>
- Kim, D., Cho, S., Tamil, L., Song, D. J. & Seo, S. (2020). Predicting asthma attacks: Effects of indoor pm concentrations on peak expiratory flow rates of asthmatic children. *IEEE Access*, 8, 8791-8797. <https://doi.org/10.1109/ACCESS.2019.2960551>
- Kim, H. C. & Ghahramani, Z. (2012). Bayesian classifier combination. In N. D. Lawrence & M. Girolami (Eds.), *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics* (Vol. 22, pp. 619–627). La Palma, Canary Islands: PMLR.
- Klawonn, F. & Angelov, P. (2006). Evolving extended naive bayes classifiers. In *Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)* (p. 643-647). <https://doi.org/10.1109/ICDMW.2006.74>
- Kolajo, T., Daramola, O. & Adebisi, A. (2019, 06). Big data stream analysis: a systematic literature review. *Journal of Big Data*, 6, 47. <https://doi.org/10.1186/s40537-019-0210-7>
- Kolter, J. Z. & Maloof, M. A. (2007). Dynamic weighted majority: An ensemble method for drifting concepts. *The Journal of Machine Learning Research*, 8(91), 2755–2790.
- Kone, A., Bronskill, S., Gruneir, A., Calzavara, A., Thavorn, K., Petrosyan, Y., ... Wodchis, W. (2015). The increasing burden and complexity of multimorbidity. *BMC public health*, 15, 415. <https://doi.org/10.1186/s12889-015-1733-2>

- Kothalawala, D., Kadalayil, L., Weiss, V., Kyyaly, M. A., Arshad, S., Holloway, J. & Rezwani, F. (2020). Prediction models for childhood asthma: A systematic review. *Pediatric Allergy and Immunology*, 31. <https://doi.org/10.1111/pai.13247>
- Kourtellis, N., De Francisci Morales, G., Bifet, A. & Murdopo, A. (2016). VHT: Vertical hoeffding tree. In *2016 IEEE International Conference on Big Data (Big Data)* (p. 915-922). <https://doi.org/10.1109/BigData.2016.7840687>
- Krempl, G., Žliobaite, I., Brzeziński, D., Hüllermeier, E., Last, M., Lemaire, V., ... Stefanowski, J. (2014). Open challenges for data stream mining research. *SIGKDD Explor. Newsl.*, 16(1), 1–10. <https://doi.org/10.1145/2674026.2674028>
- Krishnamachari, B., Estrin, D. & Wicker, S. (2002). Modelling data-centric routing in wireless sensor networks. In *IEEE Infocom* (Vol. 2, pp. 39–44).
- Kulkarni, U., Desai, P., Ahmed, J., Vadavi, J. & Yardi, A. (2007). Mobile agent based distributed data mining. In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)* (Vol. 2, p. 18-24). <https://doi.org/10.1109/ICCIMA.2007.105>
- Kumar, B. (2020). *10 Techniques to deal with Imbalanced Classes in Machine Learning*. Retrieved 02-10-2021, from <https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/>
- Kuncheva, L. & Whitaker, C. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51, 181-207. <https://doi.org/10.1023/A:1022859003006>
- Kuo, Y.-H., Chan, N. B., Leung, J. M., Meng, H., So, A. M.-C., Tsoi, K. K. & Graham, C. A. (2020). An integrated approach of machine learning and systems thinking for waiting time prediction in an emergency department. *International Journal of Medical Informatics*, 139. <https://doi.org/10.1016/j.ijmedinf.2020.104143>
- Lampe, J. B. (1993). Association between exposure to environmental tobacco smoke and exacerbation of asthma in children. *Clinical Pediatrics*, 32(9), 571-572. <https://doi.org/10.1177/000992289303200916>
- Law, Y.-N. & Zaniolo, C. (2005). An adaptive nearest neighbor classification algorithm for data streams. In *Proceedings of the 9th European Conference on European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases* (p. 108–120). Berlin, Heidelberg: Springer-Verlag.
- Leibnitz, K., Hoßfeld, T., Wakamiya, N. & Murata, M. (2007). Peer-to-peer vs. client/server: Reliability and efficiency of a content distribution service. In L. Mason, T. Drwiega & J. Yan (Eds.), *Managing Traffic Performance in Converged Networks* (pp. 1161–1172). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72990-7_99
- Levenberg, A., Simpson, E., Roberts, S. & Gottlob, G. (2013). Economic prediction using heterogeneous data streams from the world wide web. In *Scalable Decision Making: Uncertainty, Imperfection, Deliberation (SCALE), Proceedings of ECML/PKDD Workshop*.
- Li, H. & Tan, J. (2007). Body sensor network based ecg segmentation and analysis. In *2007 29th Annual International Conference of the*

- IEEE Engineering in Medicine and Biology Society* (p. 5166-5169). <https://doi.org/10.1109/IEMBS.2007.4353505>
- Li, L., Zou, B., Hu, Q., Wu, X. & Yu, D. (2013). Dynamic classifier ensemble using classification confidence. *Neurocomputing*, 99, 581-591. <https://doi.org/10.1016/j.neucom.2012.07.026>
- Li, T., Zhu, S. & Ogihara, M. (2003). A new distributed data mining model based on similarity. In *Proceedings of the 2003 ACM Symposium on Applied Computing* (p. 432-436). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/952532.952618>
- Li, Y., Bai, C. & Reddy, C. K. (2016). A distributed ensemble approach for mining healthcare data under privacy constraints. *Information Sciences*, 330, 245-259. <https://doi.org/10.1016/j.ins.2015.10.011>
- Li, Y. & Li, L. (2009). A novel split and merge em algorithm for gaussian mixture model. In *2009 Fifth International Conference on Natural Computation* (Vol. 6, p. 479-483). <https://doi.org/10.1109/ICNC.2009.625>
- Likas, A., Vlassis, N. & J. Verbeek, J. (2003). The global k-means clustering algorithm. *Pattern Recognition*, 36(2), 451-461. [https://doi.org/10.1016/S0031-3203\(02\)00060-2](https://doi.org/10.1016/S0031-3203(02)00060-2)
- Lisspers, K., Ställberg, B., Larsson, K., Janson, C., Müller, M., Łuczko, M., ... Johansson, G. (2021). Developing a short-term prediction model for asthma exacerbations from swedish primary care patients' data using machine learning - based on the arctic study. *Respiratory Medicine*, 185. <https://doi.org/10.1016/j.rmed.2021.106483>
- Liu, B., Cao, S. & He, W. (2011). Distributed data mining for e-business. *Information Technology and Management*, 12, 67-79. <https://doi.org/10.1007/s10799-011-0091-8>
- Liu, K., Kargupta, H. & Ryan, J. (2004). Distributed data mining bibliography.
- Liu, K., Kargupta, H. & Ryan, J. (2006). Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering*, 18(1), 92-106. <https://doi.org/10.1109/TKDE.2006.14>
- Liu, Q. & Wu, Y. (2012). Supervised learning. In N. M. Seel (Ed.), *Encyclopedia of the Sciences of Learning* (pp. 3243-3245). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4419-1428-6_451
- Liu, Y., Xu, Z. & Li, C. (2018). Online semi-supervised support vector machine. *Information Sciences*, 439. <https://doi.org/10.1016/j.ins.2018.01.048>
- Loo, H. R. & Marsono, M. N. (2015). Online data stream classification with incremental semi-supervised learning. In *Proceedings of the Second ACM IKDD Conference on Data Sciences* (p. 132-133). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2732587.2732614>
- Loymans, R., Honkoop, P., Termeer, E., Snoeck-Stroband, J., Assendelft, W., Schermer, T., ... Riet, G. (2016). Identifying patients at risk for severe exacerbations of asthma: Development and external validation of a multivariable prediction model. *Thorax*, 71. <https://doi.org/10.1136/thoraxjnl-2015-208138>

- Loymans, R. J., Debray, T. P., Honkoop, P. J., Termeer, E. H., Snoeck-Stroband, J. B., Schermer, T. R., ... Ter Riet, G. (2018). Exacerbations in adults with asthma: A systematic review and external validation of prediction models. *The Journal of Allergy and Clinical Immunology: In Practice*, 6(6), 1942-1952.e15. <https://doi.org/10.1016/j.jaip.2018.02.004>
- Lu, H., Frauendorfer, D., Rabbi, M., Mast, M. S., Chittaranjan, G. T., Campbell, A. T., ... Choudhury, T. (2012). Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (p. 351-360). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2370216.2370270>
- Luo, G., He, S., Stone, B., Nkoy, F. & Johnson, M. (2020). Developing a model to predict hospital encounters for asthma in asthmatic patients: Secondary analysis. *JMIR Medical Informatics*, 8. <https://doi.org/10.2196/16080>
- Luo, J., Xu, L., Jamont, J.-P., Zeng, L. & Shi, Z. (2007). Flood decision support system on agent grid: Method and implementation. *Enterprise Information Systems*, 1, 49-68. <https://doi.org/10.1080/17517570601092184>
- Luong, A. V., Nguyen, T. T., Liew, A. W.-C. & Wang, S. (2021). Heterogeneous ensemble selection for evolving data streams. *Pattern Recognition*, 112, 107743. Retrieved from <https://www.sciencedirect.com/science/article/pii/S003132032030546X> <https://doi.org/10.1016/j.patcog.2020.107743>
- Ma, X., Yang, D., Tang, S., Luo, Q., Zhang, D. & Li, S. (2004). Online mining in sensor networks. In *IFIP International Conference on Network and Parallel Computing* (pp. 544-550).
- Mahmood, A., Shi, K., Khatoon, S. & Xiao, M. (2013). Data mining techniques for wireless sensor networks: A survey. *International Journal of Distributed Sensor Networks*, 9(7). <https://doi.org/10.1155/2013/406316>
- Mallik, R., Sarda, N., Kargupta, H. & Bandyopadhyay, S. (2011). Distributed data mining for sustainable smart grids. *Proc. of ACM SustKDD*, 11, 1-6.
- Manohar, E., Jenifer, P., Nisha, M. S. & Benita, B. (2021). A collective data mining approach to predict customer behaviour. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)* (p. 1310-1316). <https://doi.org/10.1109/ICICV50876.2021.9388558>
- Mao, G. & Qiao, J. (2017). A solution for mining big data based on distributed data streams and its classifying algorithms. In Y. Tan, H. Takagi & Y. Shi (Eds.), *Data Mining and Big Data* (pp. 263-271). Springer International Publishing. https://doi.org/10.1007/978-3-319-61845-6_27
- Mao, G. & Yang, Y. (2011). A micro-cluster based ensemble approach for classifying distributed data streams. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence* (p. 753-759). <https://doi.org/10.1109/ICTAI.2011.118>
- Mao, H., Counts, S. & Bollen, J. (2011). *Predicting financial markets: Comparing survey, news, twitter and search engine data*. arXiv. <https://doi.org/10.48550/ARXIV.1112.1051>

- Marban, O., Mariscal, G. & Segovia, J. (2009). A data mining & knowledge discovery process model. In J. Ponce & A. Karahoca (Eds.), *Data Mining and Knowledge Discovery in Real Life Applications* (chap. 1). Rijeka: IntechOpen. <https://doi.org/10.5772/6438>
- Masud, M. M., Gao, J., Khan, L., Han, J. & Thuraisingham, B. (2008). A practical approach to classify evolving data streams: Training with limited amount of labeled data. In *2008 Eighth IEEE International Conference on Data Mining* (p. 929-934). <https://doi.org/10.1109/ICDM.2008.152>
- Matin, M. & Islam, M. (2012). Overview of wireless sensor network. In M. A. Matin (Ed.), *Wireless Sensor Networks* (chap. 1). Rijeka: IntechOpen. <https://doi.org/10.5772/49376>
- McConnell, S. & Skillicorn, D. B. (2004). Building predictors from vertically distributed data. In *Proceedings of the 2004 Conference of the Centre for Advanced Studies on Collaborative Research* (pp. 150–162).
- McLeod, A. I. & Bellhouse, D. R. (1983). A convenient algorithm for drawing a simple random sample. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 32(2), 182–184. <https://doi.org/10.2307/2347297>
- Mena-Torres, D. & Aguilar-Ruiz, J. S. (2014). A similarity-based approach for data stream classification. *Expert Systems with Applications*, 41(9), 4224-4234. <https://doi.org/10.1016/j.eswa.2013.12.041>
- Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., ... Hodjat, B. (2019). Chapter 15 - evolving deep neural networks. In R. Kozma, C. Alippi, Y. Choe & F. C. Morabito (Eds.), *Artificial Intelligence in the Age of Neural Networks and Brain Computing* (p. 293-312). Academic Press. <https://doi.org/10.1016/B978-0-12-815480-9.00015-3>
- Mikalef, P., Boura, M., Lekakos, G. & Krogstie, J. (2019). Big data analytics and firm performance: Findings from a mixed-method approach. *Journal of Business Research*, 98, 261-276. Retrieved from <https://www.sciencedirect.com/science/article/pii/S014829631930061X> <https://doi.org/10.1016/j.jbusres.2019.01.044>
- Miles, M. B. & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. Sage.
- Miller, M., Lee, J., Miller, D. & Wenzel, S. (2007). Recent asthma exacerbations: A key predictor of future exacerbations. *Respiratory Medicine*, 101, 481-489. <https://doi.org/10.1016/j.rmed.2006.07.005>
- Minku, L. L., White, A. P. & Yao, X. (2010). The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Transactions on Knowledge and Data Engineering*, 22(5), 730-742. <https://doi.org/10.1109/TKDE.2009.156>
- Mishra, S., Balan, R., Shibu, A. & Hota, C. (2020). Real-time probabilistic approach for traffic prediction on iot data streams. In H. Yang, K. Pasupa, A. C.-S. Leung, J. T. Kwok, J. H. Chan & I. King (Eds.), *Neural information processing* (pp. 633–641). Cham: Springer International Publishing.
- Moghadam, A. N. & Ravanmehr, R. (2017). Multi-agent distributed data mining approach for classifying meteorology data: case study on iran's synoptic weather

- stations. *International Journal of Environmental Science and Technology*, 15, 149-158. <https://doi.org/10.1007/s13762-017-1351-x>
- Mohamed, H. (2010). Data stream mining. In *Proc. of the 1st International Conference on Machine and Web Intelligence (ICMWT'2010)*, Algiers, Algeria, Oct. 2010.
- Mohanty, S., Lydia, L., Elhoseny, M., Otaibi, M. & Shankar, D. (2020). Deep learning with lstm based distributed data mining model for energy efficient wireless sensor networks. *Physical Communication*, 40. <https://doi.org/10.1016/j.phycom.2020.101097>
- Mokeddem, D. & Belbachir, H. (2009). A survey of distributed classification based ensemble data mining methods. *Journal of Applied Sciences*, 9. <https://doi.org/10.3923/jas.2009.3739.3745>
- Monteiro, C., Bessa, R., Miranda, V., Botterud, A., Wang, J., Conzelmann, G. & Porto, I. (2009). *Wind power forecasting : state-of-the-art 2009*. (Tech. Rep.). <https://doi.org/10.2172/968212>
- Muhlbaier, M. D. & Polikar, R. (2007). An ensemble approach for incremental learning in nonstationary environments. In M. Haindl, J. Kittler & F. Roli (Eds.), *Multiple Classifier Systems* (pp. 490–500). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Murphy, K. P. (2002). *Dynamic bayesian networks: Representation, inference and learning* (Unpublished doctoral dissertation). UNIVERSITY OF CALIFORNIA, BERKELEY.
- Museba, T., Nelwamondo, F. & Ouahada, K. (2021). An adaptive heterogeneous online learning ensemble classifier for nonstationary environments. *Computational Intelligence and Neuroscience*, 1-11. <https://doi.org/10.1155/2021/6669706>
- Muthukrishnan, S. (2005). Data streams: Algorithms and applications. *Theoretical Computer Science*, 1(2), 117–236. <https://doi.org/10.1561/04000000002>
- Naghavi, M., Abajobir, A., Cristiana, A., Abbas, K., Abd-Allah, F., Abera, S., ... Murray, C. (2017). Global, regional, and national age-sex specific mortality for 264 causes of death, 1980-2016: A systematic analysis for the global burden of disease study 2016. *The Lancet*, 390, 1151-1210. [https://doi.org/10.1016/S0140-6736\(17\)32152-9](https://doi.org/10.1016/S0140-6736(17)32152-9)
- Nawaz, F., Janjua, N. & Hussain, O. (2019). PERCEPTUS: Predictive complex event processing and reasoning for IoT-enabled supply chain. *Knowledge-Based Systems*. <https://doi.org/10.1016/j.knosys.2019.05.024>
- New who global air quality guidelines aim to save millions of lives from air pollution. (2021). Retrieved 30-11-2021, from <https://www.who.int/news/item/22-09-2021-new-who-global-air-quality-guidelines-aim-to-save-millions-of-lives-from-air-pollution>
- Ng, R., Sutradhar, R., Kornas, K., Wodchis, W., Sarkar, J., Fransoo, R. & Rosella, L. (2020). Development and validation of the chronic disease population risk tool (cdport) to predict incidence of adult chronic disease. *JAMA Network Open*, 3. <https://doi.org/10.1001/jamanetworkopen.2020.4669>
- Nguyen, H.-L., Woon, Y.-K., Ng, W.-K. & Wan, L. (2012). Heterogeneous ensemble for feature drifts in data streams. In P.-N. Tan, S. Chawla, C. K. Ho & J. Bailey

- (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 1–12). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-30220-6_1
- NIWA. (2006). *How is climate linked to respiratory infections in auckland?* Retrieved from <https://niwa.co.nz/publications/water-and-atmosphere/vol14-no4-december-2006/how-is-climate-linked-to-respiratory-infections-in-auckland> (28/10/2021)
- Noh, J., Sohn, J., Cho, J., Cho, S., Choi, Y., Kim, C. & Shin, D. (2016). Short-term effects of ambient air pollution on emergency department visits for asthma: An assessment of effect modification by prior allergic disease history. *Journal of Preventive Medicine and Public Health*, 49(5), 329–341. <https://doi.org/10.3961/jpmph.16.038>
- Oza, N. (2005). Online bagging and boosting. In *2005 IEEE International Conference on Systems, Man and Cybernetics* (Vol. 3, p. 2340–2345). <https://doi.org/10.1109/ICSMC.2005.1571498>
- Oza, N. (2011). *Flt Flight Simulator*. Retrieved from <https://c3.nasa.gov/dashlink/resources/294/>
- Oza, N. C. & Russell, S. (2001). Experimental comparisons of online and batch versions of bagging and boosting. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 359–364). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/502512.502565>
- Pan, B., Zheng, Y., Wilkie, D. & Shahabi, C. (2013). Crowd sensing of traffic anomalies based on human mobility and social media. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (p. 344–353). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2525314.2525343>
- Park, B.-H., Ayyagari, R. & Kargupta, H. (2001). A fourier analysis based approach to learning decision trees in a distributed environment. In *Proceedings of the 2001 SIAM International Conference on Data Mining* (pp. 1–22). Chicago, IL, USA: SIAM. <https://doi.org/10.1137/1.9781611972719.19>
- Park, B.-H. & Kargupta, H. (2003). Distributed data mining: Algorithms, systems, and applications. In *The Handbook of Data Mining* (pp. 341–358). Mahwah, United States: Lawrence Erlbaum Associates.
- Parker, B., Mustafa, A. M. & Khan, L. (2012). Novel class detection and feature via a tiered ensemble approach for stream mining. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence* (Vol. 1, pp. 1171–1178). <https://doi.org/10.1109/ICTAI.2012.168>
- Parthasarathy, S., Ghoting, A. & Otey, M. (2007, 04). A survey of distributed mining of data streams. In (Vol. 31, p. 289–307). [10.1007/978-0-387-47534-9_13](https://doi.org/10.1007/978-0-387-47534-9_13)
- Patel, S., Chamberlain, D. & Chamberlain, J. (2018). A machine-learning approach to predicting need for hospitalization for pediatric asthma exacerbation at the

- time of emergency department triage. *Academic Emergency Medicine*, 25. <https://doi.org/10.1111/acem.13655>
- Patkos, T., Plexousakis, D., Chibani, A. & Amirat, Y. (2016). An event calculus production rule system for reasoning in dynamic and uncertain domains. *Theory and Practice of Logic Programming*, 16(3), 325–352. <https://doi.org/10.1017/S1471068416000065>
- Pechenizkiy, M., Bakker, J., Žliobaitundefined, I., Ivannikov, A. & Kärkkäinen, T. (2010). Online mass flow prediction in cfb boilers with explicit detection of sudden concept drift. *SIGKDD Explor. Newsl.*, 11(2), 109–116. <https://doi.org/10.1145/1809400.1809423>
- Pintelas, P. & Livieris, I. (2020). Special issue on ensemble learning and applications. *Algorithms*, 13(6). <https://doi.org/10.3390/a13060140>
- Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3), 21–45. <https://doi.org/10.1109/MCAS.2006.1688199>
- Polikar, R., Upda, L., Upda, S. & Honavar, V. (2001). Learn++: an incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(4), 497–508. <https://doi.org/10.1109/5326.983933>
- Poosala, V., Ganti, V. & Ioannidis, Y. E. (1999). Approximate query answering using histograms. *IEEE Data Eng. Bull.*, 22(4), 5–14.
- Prasad, B. & Agarwal, S. (2017). Critical parameter analysis of vertical hoeffding tree for optimized performance using samoa. *International Journal of Machine Learning and Cybernetics*, 8. <https://doi.org/10.1007/s13042-016-0513-3>
- Prosper Marketplace, Inc. (2014). *Loan data from prosper*. Retrieved from <https://docs.google.com/document/d/1qEcwltBMlRYZT-1699-71TzInWfk4W9q5rTCSvDVMpc/pub>
- Qi, G., Sun, Y., Gao, J., Hu, Y. & Li, J. (2016). Matrix variate restricted boltzmann machine. In *2016 International Joint Conference on Neural Networks (IJCNN)* (pp. 389–395). <https://doi.org/10.1109/IJCNN.2016.7727225>
- Qi, H., Wang, T.-W. & Birdwell, J. (2004). Global principal component analysis for dimensionality reduction in distributed data mining. *Statistical Data Mining and Knowledge Discovery*, 327–342. <https://doi.org/10.1201/9780203497159.ch19>
- Qingqing Chang, J. H. (2022). Research and application of the data mining technology in economic intelligence system. *Computational Intelligence and Neuroscience*, 2022. Retrieved from <https://doi.org/10.1155/2022/6439315>
- Rahnama, A. H. A. (2014). Distributed real-time sentiment analysis for big data social streams. In *2014 International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 789–794). <https://doi.org/10.1109/CoDIT.2014.6996998>
- Ram, S., Zhang, W., Williams, M. & Pengetnze, Y. (2015). Predicting asthma-related emergency department visits using big data. *IEEE Journal of Biomedical and Health Informatics*, 19. <https://doi.org/10.1109/JBHI.2015.2404829>
- Read, J., Bifet, A., Pfahringer, B. & Holmes, G. (2012). Batch-incremental versus

- instance-incremental learning in dynamic and evolving data. In J. Hollmén, F. Klawonn & A. Tucker (Eds.), *Advances in Intelligent Data Analysis XI* (pp. 313–323). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-34156-4_29
- Recamonde-Mendoza, M. & Bazzan, A. L. (2016). Social choice in distributed classification tasks. *Inf. Sci.*, 332(C), 56–71. <https://doi.org/10.1016/j.ins.2015.11.001>
- Rehman, M. H. u., Liew, C. S. & Wah, T. Y. (2014). Frequent pattern mining in mobile devices: A feasibility study. In *Proceedings of the 6th International Conference on Information Technology and Multimedia* (pp. 351–356). <https://doi.org/10.1109/ICIMU.2014.7066658>
- Rijn, J. N., Holmes, G., Pfahringer, B. & Vanschoren, J. (2018). The online performance estimation framework: Heterogeneous ensemble learning for data streams. *Mach. Learn.*, 107(1), 149–176. <https://doi.org/10.1007/s10994-017-5686-9>
- Rokach, L. (2010). Ensemble-based classifiers. *Artif. Intell. Rev.*, 33, 1–39. <https://doi.org/10.1007/s10462-009-9124-7>
- Roughan, M. & Zhang, Y. (2006). Secure distributed data-mining and its application to large-scale network measurements. *SIGCOMM Comput. Commun. Rev.*, 36(1), 7–14. <https://doi.org/10.1145/1111322.1111326>
- Ruta, D. & Gabrys, B. (2005). Classifier selection for majority voting. *Information Fusion*, 6(1), 63–81. <https://doi.org/10.1016/j.inffus.2004.04.008>
- Rutkowski, L., Jaworski, M. & Duda, P. (2020). Basic concepts of data stream mining. In *Stream Data Mining: Algorithms and Their Probabilistic Properties* (pp. 13–33). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-13962-9_2
- Ryu, B., Yoo, S., Kim, S. & Choi, J. (2021). Thirty-day hospital readmission prediction model based on common data model with weather and air quality data. *Scientific Reports*, 11. <https://doi.org/10.1038/s41598-021-02395-9>
- Sajjad, H. P., Payberah, A. H., Rahimian, F., Vlassov, V. & Haridi, S. (2016). Boosting vertex-cut partitioning for streaming graphs. In *2016 IEEE International Congress on Big Data (BigData Congress)* (pp. 1–8). <https://doi.org/10.1109/BigDataCongress.2016.10>
- Salhi, L., Silverston, T., Yamazaki, T. & Miyoshi, T. (2019). Early detection system for gas leakage and fire in smart home using machine learning. *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 1–6. <https://doi.org/10.1109/ICCE.2019.8661990>
- Sarkar, C., Nambi, S. N. A. U., Prasad, R. V. & Biswas, A. R. (2014). A scalable distributed architecture towards unifying iot applications. *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 508–513. <https://doi.org/10.1016/10.1109/WF-IoT.2014.6803220>
- Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3). <https://doi.org/10.1016/10.1007/s42979-021-00592-x>
- Sarnovsky, M. & Kolarik, M. (2021, 04). Classification of the drifting data streams using heterogeneous diversified dynamic class-weighted ensemble. *PeerJ Computer*

- Science*, 7. <https://doi.org/10.7717/peerj-cs.459>
- Schatz, M., Cook, E. F., Joshua, A. & Petitti, D. B. (2003). Risk factors for asthma hospitalizations in a managed care organization: development of a clinical prediction rule. *The American journal of managed care*, 9(8), 538-547. (PMID =12921231)
- Schumaker, R. P. & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Trans. Inf. Syst.*, 27(2). <https://doi.org/10.1145/1462198.1462204>
- Silva, J. A., Faria, E. R., Barros, R. C., Hruschka, E. R., Carvalho, A. C. d. & Gama, J. (2013). Data stream clustering: A survey. *ACM Comput. Surv.*, 46(1). <https://doi.org/10.1145/2522968.2522981>
- Simpson, E., Roberts, S., Psorakis, I. & Smith, A. (2013). Dynamic bayesian combination of multiple imperfect classifiers. In T. V. Guy, M. Karny & D. Wolpert (Eds.), *Decision Making and Imperfection* (pp. 1–35). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-36406-8_1
- Skillicorn, D. & McConnell, S. (2008). Distributed prediction from vertically partitioned data. *Journal of Parallel and Distributed Computing*, 68(1), 16-36. <https://doi.org/10.1016/j.jpdc.2007.07.009>
- Smola, A. & Vishwanathan, S. V. N. (2008). *Introduction to machine learning*. Cambridge University Press.
- Soyiri, I. & Reidpath, D. (2012). An overview of health forecasting. *Environmental Health and Preventive Medicine*, 18(1). <https://doi.org/10.1007/s12199-012-0294-6>
- Speybroeck, N. (2012). Classification and regression trees. *International Journal of Public Health*, 57(1), 243–246. <https://doi.org/10.1007/s00038-011-0315-z>
- Spyroglou, I. I., Spöck, G., Rigas, A. G. & Paraskakis, E. (2018). Evaluation of bayesian classifiers in asthma exacerbation prediction after medication discontinuation. *BMC Research Notes*, 11(1), 1–6. <https://doi.org/10.1186/s13104-018-3621-1>
- Stahl, F., Gaber, M. M., Liu, H., Bramer, M. & Yu, P. S. (2011). Distributed classification for pocket data mining. In M. Kryszkiewicz, H. Rybinski, A. Skowron & Z. W. Raś (Eds.), *Foundations of Intelligent Systems* (pp. 336–345). Berlin, Heidelberg: Springer.
- Steyerberg, E. (2009). Applications of prediction models. In *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating* (pp. 11–31). New York, NY: Springer New York. https://doi.org/10.1007/978-0-387-77244-8_2
- Street, W. N. & Kim, Y. (2001). A streaming ensemble algorithm (SEA) for large-scale classification. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 377–382). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/502512.502568>
- Suarez-Tangil, G., Tapiador, J. E., Peris-Lopez, P. & Pastrana, S. (2015). Power-aware anomaly detection in smartphones: An analysis of on-platform versus externalized operation. *Pervasive and Mobile Computing*, 18, 137-151. <https://doi.org/10.1016/j.pmcj.2014.10.007>

- Sudharsan, B., Breslin, J. G. & Intizar Ali, M. (2021). Imbal-ol: Online machine learning from imbalanced data streams in real-world iot. In *2021 IEEE International Conference on Big Data (Big Data)* (p. 4974-4978). <https://doi.org/10.1109/BigData52589.2021.9671765>
- Suglia, S., Gryparis, A., Schwartz, J. & Wright, R. (2008). Association between traffic-related black carbon exposure and lung function among urban women. *Environmental Health Perspectives*, 116, 1333-1337. <https://doi.org/10.1289/ehp.11223>
- Sun, X. & Ansari, N. (2016). Edgeiot: Mobile edge computing for the internet of things. *IEEE Communications Magazine*, 54(12), 22-29. <https://doi.org/10.1109/MCOM.2016.1600492CM>
- Tan, G. W.-H., Lee, V.-H., Wong, C.-H. & Ooi, K. (2016). Mobile shopping: The new retailing industry in the 21st century. In *Encyclopedia of E-Commerce Development, Implementation, and Management* (p. 1448-1460). <https://doi.org/10.4018/978-1-4666-9787-4.ch102>
- Tekin, C., Canzian, L. & van der Schaar, M. (2014). Context-adaptive big data stream mining. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (p. 483-490). <https://doi.org/10.1109/ALLERTON.2014.7028494>
- Tekin, C. & van der Schaar, M. (2013). Distributed online big data classification using context information. In *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (p. 1435-1442). <https://doi.org/10.1109/Allerton.2013.6736696>
- Tennant, M., Stahl, F., Rana, O. & Gomes, J. B. (2017). Scalable real-time classification of data streams with concept drift. *Future Generation Computer Systems*, 75, 187-199. <https://doi.org/10.1016/j.future.2017.03.026>
- Thaw, A. M., Zhukova, N., Aung, T. T. & Chernokulsky, V. (2021). Data classification model for fog-enabled mobile iot systems. In H. Sharma, M. Saraswat, A. Yadav, J. H. Kim & J. C. Bansal (Eds.), *Congress on intelligent systems* (pp. 125-138). Singapore: Springer Singapore.
- Tibble, H., Tsanas, A., Horne, E., Horne, R., Mizani, M., Simpson, C. R. & Sheikh, A. (2019). Predicting asthma attacks in primary care: protocol for developing a machine learning-based prediction model. *BMJ Open*, 9(7). <https://doi.org/10.1136/bmjopen-2018-028375>
- Toonders, J. (2014). Data is the new oil of the digital economy. *Wired*. <https://www.wired.com/insights/2014/07/data-new-oil-digital-economy/>
- Tresa Paul, S., Raimond, K. & Kanaga, G. (2019, 01). An iot-enabled hadoop-based data analytics and prediction framework for a pollution-free smart-township and an asthma-free generation: Proceedings of icbdcc18. In *Advances in Big Data and Cloud Computing* (p. 577-587). https://doi.org/10.1007/978-981-13-1882-5_51
- Tsoumakas, G. & Vlahavas, I. (2009). Distributed data mining. In *Database Technologies: Concepts, Methodologies, Tools, and Applications* (pp. 157-164). IGI Global.
- Tsoumakas, G. & Vlahavas, I. P. (2002). Distributed data mining of large classifier ensembles. In *In Proceedings of Companion Volume of the Second Hellenic*

- Conference on Artificial Intelligence.*
- Tumer, K. & Ghosh, J. (2000). Robust order statistics based ensembles for distributed data mining. In H. Kargupta & P. Chan (Eds.), *Advances in Distributed and Parallel Knowledge Discovery* (p. 185-210). AAAI/MIT Press.
- Turinsky, A. & Grossman, R. (2000). A framework for finding distributed data mining strategies that are intermediate between centralized strategies and in-place strategies.
- Tyree, S., Weinberger, K. Q., Agrawal, K. & Paykin, J. (2011). Parallel boosted regression trees for web search ranking. In *Proceedings of the 20th International Conference on World Wide Web* (p. 387–396). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1963405.1963461>
- ur Rehman, M. H., Liew, C. S., Wah, T. Y., Imran, M., Salah, K., Nasser, N. & Svetinovic, D. (2021). Device-centric adaptive data stream management and offloading for analytics applications in future internet architectures. *Future Generation Computer Systems*, 114, 155-168. Retrieved from <https://www.sciencedirect.com/science/article/pii/S01677339X20303204> <https://doi.org/10.1016/j.future.2020.07.054>
- ur Rehman, M. H., Liew, C. S., Wah, T. Y. & Khan, M. K. (2017). Towards next-generation heterogeneous mobile data stream mining applications: Opportunities, challenges, and future research directions. *Journal of Network and Computer Applications*, 79, 1-24. <https://doi.org/10.1016/j.jnca.2016.11.031>
- Vaidyanathan, A., Yip, F. & Garbe, P. (2018). Developing an online tool for identifying at-risk populations to wildfire smoke hazards. *Science of The Total Environment*, 619-620, 376-383. <https://doi.org/10.1016/j.scitotenv.2017.10.270>
- van Smeden, M., Reitsma, J. B., Riley, R. D., Collins, G. S. & Moons, K. G. (2021). Clinical prediction models: diagnosis versus prognosis. *Journal of Clinical Epidemiology*, 132, 142-145. <https://doi.org/10.1016/j.jclinepi.2021.01.009>
- van Rijn, J. N., Holmes, G., Pfahringer, B. & Vanschoren, J. (2015). Having a blast: Meta-learning and heterogeneous ensembles for data streams. In *2015 IEEE International Conference on Data Mining* (pp. 1003–1008). <https://doi.org/10.1109/ICDM.2015.55>
- Varian, H. & Choi, H. (2009). Predicting the present with google trends. *Economic Record*, 88. <https://doi.org/10.2139/ssrn.1659302>
- Vijjeswarapu, D. & Natarajan, P. (2018). Multi-agent based data mining aggregation approaches using machine learning techniques. *International Journal of Engineering & Technology*, 7, 1136. <https://doi.org/10.14419/ijet.v7i3.9631>
- Villaverde, M., Aledo, D., Pérez, D. & Moreno, F. (2019). On the adaptability of ensemble methods for distributed classification systems: A comparative analysis. *International Journal of Distributed Sensor Networks*, 15(7), 1550147719865505. <https://doi.org/10.1177/1550147719865505>
- Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1), 37–57. <https://doi.org/10.1145/3147.3165>
- Vuleta, B. (2021). *How Much Data Is Created Every Day? [27 Staggering Stats]*. Retrieved 05-12-2021, from <https://seedscientific.com/>

- how-much-data-is-created-every-day/
- Wang, F., Hu, L., Zhou, J., Hu, J. & Zhao, K. (2017). A semantics-based approach to multi-source heterogeneous information fusion in the internet of things. *Soft Comput.*, 21(8), 2005–2013. <https://doi.org/10.1007/s00500-015-1899-7>
- Wang, H., Fan, W., Yu, P. S. & Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 226–235). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/956750.956778>
- Wang, L. (2017). Heterogeneous data and big data analytics. *Automatic Control and Information Sciences*, 3(1), 8–15. <https://doi.org/10.12691/acis-3-1-3>
- Wang, T., Zheng, Z., Rehmani, M. H., Yao, S. & Huo, Z. (2019). Privacy preservation in big data from the communication perspective—a survey. *IEEE Communications Surveys & Tutorials*, 21(1), 753–778. <https://doi.org/10.1109/COMST.2018.2865107>
- Wang, W. & Zhang, M. (2020). Data stream adaptive partitioning of sliding window based on gaussian restricted boltzmann machine. In Q. Liang, W. Wang, J. Mu, X. Liu, Z. Na & B. Chen (Eds.), *Artificial Intelligence in China* (pp. 220–228). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-15-0187-6_25
- Wares, S., Isaacs, J. & Elyan, E. (2019). Data stream mining: methods and challenges for handling concept drift. *SN Applied Sciences*, 1. <https://doi.org/10.1007/s42452-019-1433-0>
- Webb, G. I. (2010). Naïve bayes. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 713–714). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-30164-8_576
- Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L. & Petitjean, F. (2016). Characterizing concept drift. *Data Min. Knowl. Discov.*, 30(4), 964–994. <https://doi.org/10.1007/s10618-015-0448-4>
- Weinberg, A. & Last, M. (2017). Interpretable decision-tree induction in a big data parallel framework. *International Journal of Applied Mathematics and Computer Science*, 27. <https://doi.org/10.1515/amcs-2017-0051>
- Whitt, W. & Zhang, X. (2019). Forecasting arrivals and occupancy levels in an emergency department. *Operations Research for Health Care*, 21, 1–18. <https://doi.org/10.1016/j.orhc.2019.01.002>
- Widmer, G. & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Mach. Learn.*, 23(1), 69–101. <https://doi.org/10.1023/A:1018046501280>
- Witton, B. (2019). 'We are worried' — doctors fear patients will die waiting in ambulances this winter, unable to access overcrowded emergency departments. Retrieved 09-01-2022, from <https://www.stuff.co.nz/national/health/124993556/we-are-worried--doctors-fear-patients-will-die-waiting-in-ambulances-this-winter-unable-to-access-overcrowded-emergency-departments>
- Woods, K., Kegelmeyer, W. & Bowyer, K. (1997). Combination of multiple classifiers

- using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), 405-410. <https://doi.org/10.1109/34.588027>
- Yang, Q. & Wu, X. (2006). 10 challenging problems in data mining research. *Int. J. Inf. Technol. Decis. Mak.*, 5(4), 597-604. <https://doi.org/10.1142/S0219622006002258>
- Yang, Y., Chen, Z., Yan, J., Xiong, Z., Zhang, J., Tu, Y. & Yuan, H. (2019). Multi-source heterogeneous information fusion of power assets based on knowledge graph. In *2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)* (p. 213-218). <https://doi.org/10.1109/SOLI48380.2019.8955005>
- Yasumoto, K., Yamaguchi, H. & Shigeno, H. (2016). Survey of real-time processing technologies of iot data streams. *Journal of Information Processing*, 24, 195-202. <https://doi.org/10.2197/ipsjjip.24.195>
- Ye, L., Jiang, Z., Yao, L., Wang, X., Yiu, S. & Huang, Z. (2019). Outsourced privacy-preserving C4.5 decision tree algorithm over horizontally and vertically partitioned dataset among multiple parties. *Cluster Computing*, 22. <https://doi.org/10.1007/s10586-017-1019-9>
- Yi, X., Zhang, J., Wang, Z., Li, T. & Zheng, Y. (2018). Deep distributed fusion network for air quality prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (p. 965-973). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3219819.3219822>
- Zahran, H., Bailey, C., Damon, S., Garbe, P. & Breysse, P. (2018). Vital signs : Asthma in children — United States, 2001–2016. *MMWR. Morbidity and Mortality Weekly Report*, 67. <https://doi.org/10.15585/mmwr.mm6705e1>
- Zaidi, S., Abidi, S. & Manickam, S. (2002). Distributed data mining from heterogeneous healthcare data repositories: towards an intelligent agent-based framework. In *Proceedings of 15th IEEE Symposium on Computer-Based Medical Systems (CBMS 2002)* (p. 339-342). <https://doi.org/10.1109/CBMS.2002.1011401>
- Zeng, L., Li, L., Duan, L., Lu, K., Shi, Z., Wang, M., ... Luo, P. (2012). Distributed data mining: A survey. *Information Technology and Management*, 13. <https://doi.org/10.1007/s10799-012-0124-y>
- Zeng, L., Xu, L., Shi, Z., Wang, M. & Wu, W. (2007). Distributed computing environment: Approaches and applications. In *2007 IEEE International Conference on Systems, Man and Cybernetics* (p. 3240-3244). <https://doi.org/10.1109/ICSMC.2007.4413876>
- Zhang, L., Xie, Y., Xidao, L. & Zhang, X. (2018). Multi-source heterogeneous data fusion. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)* (p. 47-51). <https://doi.org/10.1109/ICAIBD.2018.8396165>
- Zhang, M. & Wang, W. (2021). Weighted adaptive partition for heterogeneous iot data stream. *IEEE Internet of Things Journal*, 8(20), 15240-15248. <https://doi.org/10.1109/JIOT.2020.3045726>
- Zhang, P., Gao, B. J., Zhu, X. & Guo, L. (2011). Enabling fast lazy learning for data streams. In *2011 IEEE 11th International Conference on Data Mining*

- (p. 932-941). <https://doi.org/10.1109/ICDM.2011.63>
- Zhao, L., Ni, L., Hu, S., Chen, Y., Zhou, P., Xiao, F. & Wu, L. (2018). InPrivate Digging: Enabling tree-based distributed data mining with differential privacy. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications* (p. 2087-2095). <https://doi.org/10.1109/INFOCOM.2018.8486352>
- Zheng, Y. (2015). Methodologies for cross-domain data fusion: An overview. *IEEE Transactions on Big Data*, 1(1). <https://doi.org/10.1109/TBDATA.2015.2465959>
- Zhou, Y., Tian, Y., Liu, F., Liu, J. & Zhu, Y. (2019). Privacy preserving distributed data mining based on secure multi-party computation. In *2019 IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT)* (p. 173-178). <https://doi.org/10.1109/ICAIT.2019.8935900>
- Zhou, Z., Zheng, W.-S., Hu, J.-F., Xu, Y. & You, J. (2016). One-pass online learning: A local approach. *Pattern Recognition*, 51, 346-357. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0031320315003180> <https://doi.org/10.1016/j.patcog.2015.09.003>
- Ziffer, G., Bernardo, A., Della Valle, E., Cerqueira, V. & Bifet, A. (2022, 12). Towards time-evolving analytics: Online learning for time-dependent evolving data streams. *Data Science*, 1-16. <https://doi.org/10.3233/DS-220057>
- Zliobaite, I., Bifet, A., Read, J., Pfahringer, B. & Holmes, G. (2015). Evaluation methods and decision theory for classification of streaming data with temporal dependence. *Machine Learning*, 98(3), 455-482. <https://doi.org/10.1007/s10994-014-5441-4>

Appendix A

Glossary

Aggregator A site that is responsible for generating the final classification for each record by aggregating the classification results from all the primary and trouble sites.

Agreement The portion of trouble records processed at the trouble site because it were declared trouble by all the source sites out of the total records transmitted to it by the source sites.

Classification A supervised predictive modelling concept in which a class label is predicted for an input data instance.

Combination A mathematical concept that computes the total number of possible selection from a larger set of distinct items, when the order of selection doesn't matter. It basically demonstrates the distinct subsets from the larger set.

Concept Drift Changes that may occur abruptly or gradually in the underlying relationships in a data stream.

Conditional probability The likelihood or probability that an event might occur based on the assumption that another (previous) event has already occurred.

Confidence Threshold A threshold value which identifies the trouble records by comparing the confidence of classifying a record fragment to the confidence threshold.

Cross-term A relationship or pattern that may exist between the record fragments (or local view) seen across the distributed sites which can only be detected by combining the views from those sites.

DDM Distributed Data Mining; a category of procedures used for mining data which is distributed across multiple geographical sites.

Feature Scaling A technique used to standardize the range of independent variables or feature set of data.

Grace Period A parameter used for creation of Hoeffding Tree that specifies number of instances a leaf node should see between different split attempts.

HDSM Hierarchical Distributed Stream Miner; an architecture used as foundation work for mining distributed data streams that uses hierarchy of trouble sites.

Horizontal Distribution Also known as homogeneous data partitions, here data is distributed across the diverse sites such that each site consists of same feature set but comprises of distinctive set of records.

Lagged Effect It is time lag between one event and other related event, it signifies the delayed reaction of a dependent variable when changes are witnessed in an independent variable.

Minimized Tree A tree created in MTDM using the least set of records from all the primary sites.

MTDM Minimized Tree for Distributed Mining; A methodology that proposes generation and parsing of a minimized tree for learning the relationship between the vertically distributed data.

Outlier An observation that is substantially different from other data points.

Pearson Correlation Coefficient A computation that denotes the intensity of linear relationship between two variables.

PODSM Performance Optimizer in Distributed Stream Mining; An algorithm which is targeted towards decreasing the resources used for mining vertically distributed

data streams.

Prequential evaluation A technique used for measuring accuracy in online classification scenario where the classifier classifies the record first and then immediately uses it for training.

Primary Site A site used in HDSM, PODSM, MTDM that signifies vertical fragment of records or heterogeneous sources.

Record Fragment Particularly referred to as vertical record fragment, it is a part of record comprising of only a subset of features of original record.

Source Site A trouble site is formed by merging two source sites; which can be either a primary site or other trouble site of lower order and may forward their trouble records to the formed trouble site.

Split Confidence A parameter used in Hoeffding Tree generation which denotes the amount of error allowed in a split decision.

Trouble Factor A configuration parameter set for each trouble site that identifies the fraction of the data stream that would be transmitted in the form of trouble records to that trouble site.

Trouble Record A record fragment that is classified with low confidence and as a result is transmitted from a source site to a trouble site.

Trouble Site A site used in HDSM, PODSM and MTDM algorithms which receives the record fragment from the source sites and then combines them to generate its classification model. The model thus obtained has the potential to capture the cross-terms between its source sites.

Trouble Site Blacklist Monitor A monitor that detects the occurrence of a concept drift.

Trouble Site Creation Monitor A monitor that is responsible for keeping an eye over the sets of source sites and regulates the creation of a new trouble site based on the agreement among the source sites.

Trouble Site Removal Monitors A monitor that decides whether to remove and black-list a trouble site which depends on two conditions; either due to less involvement in the final classification or the classification accuracy is low.

Trouble Site Order Depicts the position of a trouble site in the trouble site hierarchy structure, as such the primary sites are of order 0 and the order of a trouble site is one greater than the maximum order of its source sites.

Triggers Activators that exacerbate asthma conditions in asthma patients such as cold temperature.

Unconfident Site A site that classifies a record with confidence less than the threshold value, for that particular record, the site turns out to be unconfident.

Vertical Distribution/Partitions The dissemination of data among diverse sites in such a way that for the same set of records each site captures different view with its distinctive feature set.

Appendix B

Ethics

B.1 Ethics Approval Letter

For validating the proficiency of the proposed algorithms, a case study that could predict peak demands for asthma-related ED visits and hospitalizations was chosen. For doing experimentation on this case study, the initial requirement was to acquire daily number of asthma-related ED and hospitalizations, for which an ethics application was made to Auckland Health Research Ethics Committee (AHREC). However, an ethics application and the Participation Information sheet was also submitted to AUT to check if ethics approval was required from AUT's end. As the study included De-identified data, the Ethics Committee of AUT decided no requirement for ethics approval.

AUCKLAND HEALTH RESEARCH ETHICS COMMITTEE (AHREC)

11/05/2021

Dr Amy Chan

School of Pharmacy

Re: Application for Ethics Approval (Our Ref. AH22149): Approved

The Committee considered your application for ethics approval for the study entitled "**A Practical Tool to Predict Asthma-Related Emergency Department Visits and Admissions at Population Level**".

We are pleased to inform you that ethics approval has been granted.

The expiry date for this approval is **11/05/2024**.

Amendments to the approved project:

Should you need to make any changes to the approved project, please follow the steps below:

- Send a request to the AHREC Administrators to unlock the application form (using the Notification tab in the Ethics RM form).
- Make all changes to the relevant sections of the application form and attach revised documents (as appropriate).
- Change the Application Type to "Amendment request" in Section L.
- Add a summary of the changes requested in the text box.
- Submit the amendment request (PI/Supervisors only to submit the form).

If the project changes significantly, you are required to submit a new application.

Funded projects: If you received funding for this project, please provide this approval letter to your local Faculty Research Project Coordinator (RPC) or Research Project Manager (RPM) so that the approval can be notified via a Service Request to the Research Operations Centre (ROC) for activation of the grant.

The Chair and the members of AHREC would be happy to discuss general matters relating to ethics approvals. If you wish to do so, please contact the AHREC Ethics Administrators at ahrec@auckland.ac.nz in the first instance.

Additional information:

- Do not forget to fill in the 'approval wording' on the PISs, CFs and/or advertisements, using the date of this approval and the reference number, before you use the documents or send them out to your participants.

All communications with the AHREC regarding this application should indicate this reference number: **AH22149**.

AHREC Administrators

Auckland Health Research Ethics Committee

Appendix C

Source Code

Source code for the algorithms proposed in this study have been made available in GitHub repositories. For experimental evaluation included in Chapters 3, 4, 5, and 6, separate repositories were created. The details of all the repositories have been listed below:

1. *Chapter 3* - Experiments and the data preparation files for state-of-the-art work and PODSM has been made available at https://github.com/rashi-bhalla/PODSM_state.git.
2. *Chapter 4* - The experiments and the data preparation files for state-of-the-art work and MTDM can be accessed at https://github.com/rashi-bhalla/MTDM_state.git.
3. *Chapter 5* - The experiments for all the methodologies discussed in this chapter can be accessed at https://github.com/rashi-bhalla/Pred_model_Asthma.git. However, due to ethical constraints, the asthma hospital data file has not been uploaded into the repository but can be made available on request.
4. *Chapter 6* - The experiments for validating PODSM and MTDM on asthma dataset has been made available at <https://github.com/rashi-bhalla/>

`PODSM_on_asthma.git` and https://github.com/rashi-bhalla/MTDM_on_asthma.git. These repository also contains state-of-the-art experiments. The hospital data file has not included as mentioned earlier due to ethical reasons.