Republic of Iraq

Ministry of Higher Education and

Scientific Research

University of Babylon

Collage of Information Technology

Stream mining for Intrusion Detection Using Metaheuristics and Classification Technique

A Thesis

Submitted to the Council of the College of Information Technology at University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology / software

By

Haneen Farhan Kahdam sadkan

Supervised by

Asst. Prof. Dr. Mehdi Ebady Manaa

2021 A.D **1443 A.H**


بسم الله الرحمن الرحيم
يرفع الله الذين آمنوا منكم
والذين أوتوا العلم درجات والله
بما تعملون خبير
صدق الله العظيم
سورة المجادلة . آية 11

Supervisor Certification

I certify that this thesis was prepared under my supervision at the Department of Information Networks / College of Information Technology / University of Babylon, by **Haneen Farhan Kahdam** as a partial fulfillment of the requirements for the degree of **Master in Information Technology**.

Signature:

Name: **Asst. Prof. Dr. Mehdi Ebady Manaa**

Title: **Assistant Professor**

Date: / / 2022

The Head of the Department Certification

In view of the available recommendation, we forward this thesis for debate by the examining committee.

Signature:

Name: **Asst. Prof. Dr. Ahmed saleem**

Title: **Professor**

Date: / / 2022

Certification of the Examination Committee

We hereby certify that we have studied the thesis entitled (**STREAM MINING FOR INTRUSION DETECTION USING METAHEURISTIC AND CLASSIFICATION TECHNIQUE**) presented by the student (**Haneen Farhan Kahdam**) and examined her in its content and what is related to it, and that, in our opinion, it is adequate with (**Very Good**) standing as a thesis for the degree of Master in Information Technology- Software.

Signature:

Name: Nidaa A. Abbass

Title: Prof. Dr.

Date: / / 2022

(Chairman)

Signature:

Name: Ruaa Safaa

Title: Dr.

Date: / / 2022

(Member)

Signature:

Name: Haider Nasar khrabit

Title: Asst. Prof.

Date: / / 2022

(Member)

Signature:

Name: Mehdi Ebadi Manaa

Title: Asst. Prof.

Date: / / 2022

(Member and Supervisor)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:

Name: Hussein Atiyah. Lafta

Title: Prof.

Date: / / 2022

(Dean of the College of Information Technology)

Declaration

I hereby declare that this thesis, submitted to the University of Babylon in partial fulfillment of the requirement for the degree of Master in Information Technology \Software, has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose source is appropriately cited in the references.

Signature:

Name: Haneen Farhan Kahdam

Title: Student

Date: / / 2022

Dedication

I dedicte this thesis

To all martyrs of Islamic faith

To all defenders of the holy shrines

To the victory leaders

To the media martyr ahmed mhana al-lami

To my lovely father, mother and all the family

To my husband,, and friends

To my supervisor

The researcher

Acknowledgements

Before anything, I would like to introduce the greatest praise and thanks to the source of tender, Allah and the 12'Th imam Mahdi(god hasten his reappearance) for helping me to overcome all the challenges and difficulties which I faced at my work and finish this work.

I would like to thank my supervisor, Dr. Mehdi Ebady Manaa, for the encouragement and advice he has provided throughout my studying period. Also I would also like to thank all the members of staff at Babylon University who helped me.

I must express my gratitude to those who be as our defenders which protect us from ISIS and every dangerous surrounding us till now.

Finally, I would like to express my thanks to my husband, family and all friends for supporting me directly or indirectly to accomplish my work.

The researcher

List of Figures		
Figure number	**Figure name**	**Page number**
Figure 1.1	General data stream mining procedure	1
Figure 2.1	Stream clustering technique	14
Figure 2.2	Stream classification technique	16
Figure 2.3	IoT key stages	19
Figure 2.4	IDS (intrusion detection system)	20
Figure 2.5	Statistical data type	21
Figure 2.6	Main steps of using neural network classifier with the input selected chunks data	24
Figure 2.7	Dataset has 2 features (X, Y) and two color (blue, green)	25
Figure 2.8	Main steps of using SVM classifier	25
Figure 2.9	Concept of the silhouette index	29
Figure 2.10	Confusion matrix	32
Figure 3.1	Main stages of the proposed system	36
Figure 3.2	Proposed system of stream mining for IDS data	37
Figure 4.1	Used machine learning	50
Figure 4.2	Silhouette index values to evaluate PSO chunks	52

Table of Contents

List of tables

List of Abbreviations	
ABC	Artificial Bee Colony Algorithm
ANN	Artificial neural network
CFS-BA	Correlation-based feature selection-Bat algorithm
DA	Dragonfly Algorithm
EBat	enhanced Bat algorithm
FFA	firefly optimization algorithm
GA	genetic algorithm
GWO	grey wolf optimizer
IDS	Intrusion detection system
IoT	Internet of Things
LNNLS-KH	enhanced krill swarm algorithm which it based on linear nearest neighbor lasso step
MI	mutual information
NIDS	Network Intrusion detection system
NV	Naïve Bayes
PSO	Particle Swarm Optimization
RF and PA	Random Forest and Penalizing Attributes (Forest PA) algorithms
SEKS	search economics with k-means with SVM
SVM	Support Vector Machine

Abstract

Data stream mining comes to play an integral role in real-time applications. The main sources of the data stream flows are sensors, multimedia and social media. It contains remote sensors, stock markets, tweets, and video surveillance systems. The data stream has distinctive characteristics which include its high speed and very huge volume, in addition to its ability to change over time. It has many challenges, one of which is the concept drift that happens due to the continuous property of data stream. Traditional data mining techniques could not deal with or mine this big, rapid data.

The aim of this work is to classify sensor data by generate a powerful system which could make the classification task more accurate. This could be done by using metaheuristics with the classification technique.

The metaheuristic approach is used to build the balanced chunks of data using Particle Swarm Optimization (PSO) to obtain a better classification accuracy. The Meta model formed by the metaheuristic and classification techniques shows an improvement in accuracy performance at a high rate, as compared to non-metaheuristic models. Multiple classifiers have been chosen based on mathematical equation for each chunk to select the optimal one which gives the optimal results with high accuracy.

The obtained results showed a good performance in terms of classification accuracy for the neural network classifier 90% and low positive rate.

Chapter One

General Introduction

1.1 Introduction

The data stream comes from various sources that carry big data characteristics. Volume, velocity and veracity are considered the main issues in stream data. Data stream mining began to have an essential role in real-time applications, whereby the data is obtained through different sources such as remote sensors, scientific processes, stock markets, online transactions, tweets, internet traffic, and video surveillance systems. Stream mining is the process of producing knowledge from a continuous stream of big data. This data arrives rapidly and in an extremely large size. Its continuous property implies that its values change over time in a very fast form. Besides, another challenge for data stream mining is its need for unlimited storage size [1][2]. Figure 1.1 explains the general idea for stream mining. Machine learning and data mining are used to generate a model or pattern and to discover knowledge from a large dataset.

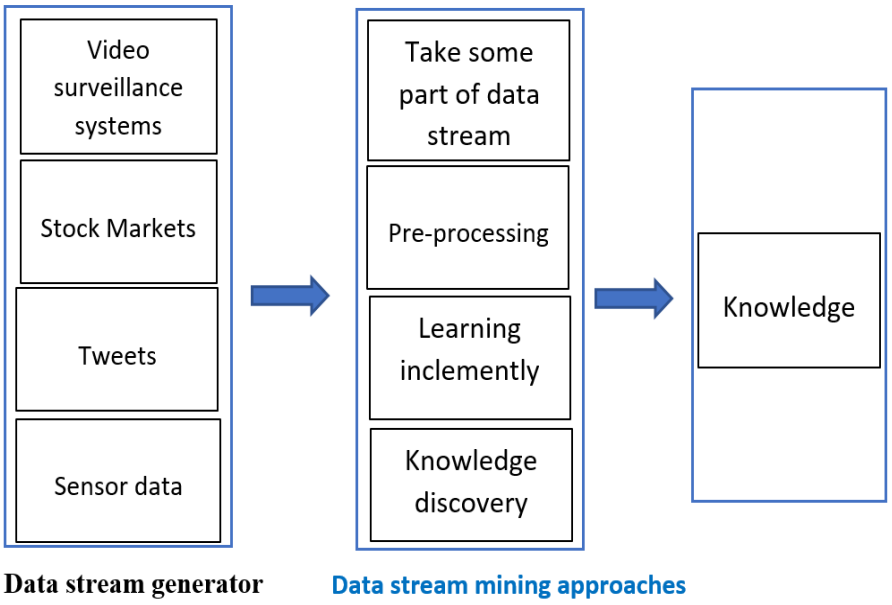

Figure 1.1: General data stream mining procedure [1]

Generally, data mining can be defined as the process of mining or delving deep into data of different forms for discovering patterns and gathering knowledge about these patterns. The large datasets are first sorted, followed by recognizing patterns and building relationships for analyzing the data and solving the issues in data mining processes [3].

Traditional data mining techniques could not deal or mine the data stream because of to its huge volume and fast changing over time (the continues property) [4]. A powerful system is applied using several classification and metaheuristic techniques to address these problems in many aspect fields.

The common thought for classification is to forecast the objective class for the given dataset by analyzing the training dataset. This means that the classification is the issue regarding the determination of new observations according to the set of categories (sub-population) to which they belong [3][5].

Metaheuristic methods involve different ways for solving the optimization problems. These methods gain the structure knowledge of the problem by employing all the information extracted from the potential solution that is recently estimated, which is then utilized to create better solutions. In the various stages of the optimization process, all metaheuristic methods rely on the use of random numbers (probabilistic approaches) [6][7]. The used technique in the thesis was Particle Swarm optimization (PSO) because of PSO offers the advantages of a quick search speed, memory, a small number of parameters, and a simple structure, as well as being easier to implement at the validation stage. In addition, it has flaws such as an imbalance between global and local search, low convergence precision, easy fall into optimal local solution, and weak robustness [8].

Data stream has many challenges to deal with, including the following points:

1. The large size of stream data and the high speed cause a different problem with the storage management, whereby the summarized techniques must be used to deal with this problem.
2. When analyzing the dataset, some of its values might rapidly change overtime. This requires a new better technique to deal with adjusted data values.
3. This data has to be mined within a limited period of time and with limited storage. However, the data stream may be different from static data, as illustrated in Table (1) below.

Table 1.1: Comparison between Data mining and stream mining [8]

Characteristics/domains	**Traditional data mining**	**Data stream mining**
No. of passes	Multiple	Single
Processing time	Unlimited	Restricted
Memory usage	Unlimited	Restricted
result Type	Accurate	Approximate
Concept	Static	Evolving
Distributed	No	Yes

Data streaming has distinct properties including its unlimited size, sequential order, and dynamic change which requires more than traditional processing. Stream mining differs from data mining, where stream mining has to treat all the data sequentially with a restricted amount of memory and restricted processing time. In

addition, it needs to find out and process all the changes in values after a certain period of time because of the continuous property of the data stream. Unlike stream mining, data mining can treat the static data at any time with unlimited memory and time because of the fixed nature of the data [2].

This thesis try to treat the problem of intrusion detection system which gathered by IoT (Internet of Things) devices. The intrusion detection system (IDS) is a system that monitors network traffic for suspicious activity and issues alerts when such activity is discovered. It is a software application that scans a network or a system for harmful activity or policy breaching [9].

1.2 Problem Statement

Data streams are big data that arrive rapidly within different period of time. It is created by both online and offline resources. Having unstable results of analysis, it is rather difficult to mine iteratively as compared to the static data, because it requires a large storage and advanced techniques. In addition, the size of data streams are unbounded and not known, which might make the user confused about the size of dataset at the beginning. Since data streams differ in quality, some of the data set values change within a certain period of time of analysis [1] [2][10]. The splitting of the dataset would balance the data, which make the classification more accurate. The director for the splitting task could be the silhouette index to evaluate the degree of the similarity between the split data.

1.3 Research Aims

The aim of this work is to enhance the classification accuracy. A hybrid technique is proposed which consists of metaheuristic and classification techniques. The objectives can divided into the following:

1. Creating a proactive system which classifies the data stream into a more accurate format.
2. Optimizing the data by using Particle Swarm Optimization (PSO) to evaluate two chunks from the dataset.
3. Finding out the optimal classifier among three classifiers (Neural Network, SVM, Naïve bays).

1.4 Scope of the Work

This scope of this work is application of real-time in stream mining for network sensor data to detect the intrusion that happened in the IoT device logs dataset [11]. It must to build a proactive model to classify the data stream with higher accuracy by utilize the metaheuristic technique which represented by Particle Swarm Optimization (PSO).

1.5 Contribution of the Research

The main contribution of this research is the use of Particle Swarm Optimization (PSO) for partitioning the dataset into two chunks. These chunks contain the optimum of the dataset which are evaluated the chunks using the threshold and silhouette index. This is followed by training one of them with the test sample using three classifiers (neural network, SVM, naïve Bayes). The using of PSO algorithm to split the data contributed to increasing the accuracy of the classification task to be more efficiency.

1.6 Related Works

This part of the thesis summarizes the previous related works that are associated with using metaheuristics with different methods, and the classification

of intrusion detection task, most of these works had used the metaheuristic technique in feature selection task.

In the last few years, stream data mining gradually became the focus of attention, where the majority of recent researches try to build a powerful system to mine this continuous large data using machine learning techniques. One of these studies is [12]. Intrusion detection system (IDS) is a system utilized for detecting and stopping the abnormal status in networks by using values of the feature from network packet capture mechanism for classifications whereby the status is either abnormal or not. In this work, a procedure has been followed which combines clustering, classification, and metaheuristic algorithms as a classification algorithm for this system, named Search Economics with K-means with SVM (SEKS). The accuracy of the proposed system from 80.37% up to 99.11%. This study tried to combine unsupervised learning algorithms with the classification algorithm of IDS to improve the accuracy rate of attack detection but it had been worked with some dataset given a high accuracy rate and start to low with another one.

The research work in [13] presents a meta-heuristic Bayesian network classification to detect intrusions. It was proposed to defeat the issue of the intrusion-based data packets in the network. The procedures of this method included pre-processing, feature selecting, feature optimizing, and classification. The data packet were classified effectively into normal or abnormal using the enhanced detecting capabilities which are determined by the optimization methods. The result of this method obtained an efficiency of 82.99%. This framework performs two key processes: preprocessing, optimization, and the classification method but it had been had a low accuracy rate compared to ours.

The work in [14] is a new method that has been proposed to detect the intrusion in networks. They used an enhanced Bat algorithm (EBat) to train the artificial neural network. The purpose of their work is to make the accuracy of the classification (where the network traffic is normal or malicious) higher. The use of the developed EBat algorithm was to select appropriate weights and biases. Then, the neural network is employed utilize the weights and biases to improve the detection of intrusions. The proposed method is applied to four dataset and the accuracy rates were 94.47%, 97.20%, 98.90%, and 97.54%. This research developed and implemented an IDS framework based on an MLP trained with the EBat method. It got a high accuracy rates but the optimizing method was enhanced not pure.

In [15], the authors propose a PSO and feature selection for intrusion detection systems. By applying random forest, they were able to decrease the irrelevant attribute as feature selection. After selecting the features, multiple classifiers have been performed to create a comparative study. The use of PSO algorithm was applied to the selected feature to enhance and maintain the false detection rate. The results obtained by this method are 99.32% for efficiency and 99.26% for the detection rate. In this study the classifier chooses the top 10 features based on their relevance and discards the rest but the proposed approach got a 87.83 in the testing set.

The work in [16], is based on feature selection and ensemble learning approaches. They propose a new intrusion detection method. To begin, the dimensions were reduced by employing the heuristic CFS-BA algorithm for selecting the best sub-set according to feature correlation. Next, the author then introduces the ab integrate of C4.5, Random Forest (RF), and Forest by Penalizing Attributes (Forest PA) algorithms as ensemble methods. At last, for attack recognition, the voting mechanism is applied to aggregate the probability distributions of base learners. The accuracy rates for this method were 99.81%, and

99.8%. A vote classifier, which is an ensemble of classifiers method, and an efficient ML-based IDS employing a metaheuristic optimization algorithm based feature selection strategy.

The authors in [17], introduced the Combination Approach of Two Metaheuristic Algorithm for Optimal Feature Selection. Two metaheuristics methods (integration of whale optimization and flower pollination algorithms) have been proposed to gain optimal feature selection. The proposed method is applied to detect the email spam, which provides powerful results in terms of accuracy for the classification task. The feature was chosen using the binary version of the whale optimization techniques and flower pollination in this study.

Moreover in [18], a Feature Selection Model is presented for NIDS using PSO, GWO, FFA and GA algorithms. This paper proposes a feature selection to improve the NIDS (network intrusion detection system). Four algorithms have been utilized by this model, namely PSO, Grey Wolf Optimizer (GWO), Fire-fly Optimization (FFA) and Genetic Algorithm (GA). It involves deploying wrapper-based methods for all algorithms and filtering-based methods for MI (mutual information). This results in thirteen sets of rules and whose accuracy is checked using the Support Vector Machine and J48 ML classifiers. The accuracy had been increased of the NIDS system by minimizing the number of features. The accuracy rate for J48 gave 79.175% to 90.484% and SVM result 79.077%–90.119%. It was challenging to choose subgroup features. When the feature's dimensionality is large, it's impossible to manage efficiently.

The researchers in [19] propose a new method to detect intrusions, which makes use of the Artificial Bee Colony (ABC) and Dragonfly Algorithms (DA) to train the artificial neural network (ANN). Firstly the best biases and weights are

chosen by using the hybrid (ABC) and (DA). Then, the neural network was retrained with these new values. This method showed a good performance, whose results for the four data sets are (94.4, 91.14, 88.7, and 91.7). This study only tested the model with intrusion detection datasets that did not include an adequate feature selection technique.

The work in [20] proposes a feature selection model to detect intrusions intrusion detection by utilizing enhanced krill swarm algorithm. It depends on linear nearest neighbor lasso step (LNNLS-KH), the number of chosen features and classification precision are presented for evaluating the fitness function of the algorithm, along with the physical dissemination movement of the krill individuals which are changed by a non-linear strategy. To find the optimal global solution, the LNNLS was applied on the updated krill flock positioning. The results showed that the classification accuracies expanded by 10.03% and 5.39%, and the detection rates expanded by 8.63% and 5.45%. The time required to detect intrusions diminished by 12.41% and 4.03% to normal. The LNNLS-KH algorithm's initialization has a certain amount of unpredictability. Despite the fact that the proposed algorithm has a promising track record.

In the last work [21], a new model was proposed for detecting the intrusion utilized the PSO for selecting features PMSO. This method treats the issue of falling the PSO in the local optimum which would give it an increase exploration rate and at the same time minimizes the probability of Stagnation. The method results show that the classification accuracy increase by 2% rather than PSO and BAT optimization. PMSO strategy got increased by 2% and this was the weakness compared to ours.

Table 1.2: Summary of related works

Authors	Dataset	Methods	Results
Z.-H. Chen and C.-W. Tsai	NSL-KDD, CIDDS-001, GPRS-WPA2, and synthetic dataset	named search economics with k-means with SVM (SEKS)	80.37% up to 99.11%.
M.K. Prasath and B.Perumal	NSL-KDD	Bayesian network classification	82.99%
W. A. H. M. Ghanem and A. Jantan	KDD Cup 99	enhanced Bat algorithm (EBat) and neural network	94.47%, 97.20%, 98.90%, and 97.54%
N. Kunhare, R. Tiwari, and J. Dhar	NSL-KDD	Particle swarm optimization and feature selection	99.26%
Y. Zhou, G. Cheng, S. Jiang, and M. Dai	NSL-KDD, AWID and CIC-IDS2017	CFS-BA algorithm and ensemble method	99.81%, and 99.8%
H. Mohmmadzadeh	UCI	whale optimization and flower pollination algorithms	92.16%
O. Almomani	UNSW-NB15	PSO, GWO, FFA and GA Algorithms with SVM and J48 ML	J48 79.175% to 90.484% and SVM 79.077%–90.119%

W. A. H. M. Ghanem, A. Jantan	KDD CUP 99, NSL-KDD, ISCX 2012, and UNSW-NB15	Artificial Bee Colony (ABC) and Dragonfly Algorithms (DA) with neural network	94.4, 91.14, 88.7, and 91.7
X. Li, P. Yi, W. Wei, Y. Jiang, and L. Tian	NSL-KDD	enhanced krill swarm based on linear nearest neighbor lasso step (LNNLS-KH)	expanded by 10.03% and 5.39%
S. S. Alkafagi and R. M. Almuttairi	NSL-KDD	PSO and classification	increase by 2%

1.7 Thesis Outlines

The remaining chapters of this thesis are ordered in the following sequence: Chapter Two present the theoretical background, as it introduces the stream mining with the metaheuristics and classification techniques. Chapter Three shows the proposed model for classifying the network sensor data. Chapter Four shows the main results of the proposed system using classification and metaheuristics techniques. Chapter Five states the conclusions that have been drawn after conducting this study, followed by some suggested works to be carried out in the future.

Chapter Two

THEORETICAL BACKGROUND

2.1 Introduction

This chapter presents the theoretical background, as it introduces the stream mining with metaheuristics and classification techniques. At the beginning, Section 2.2 contains an introduction about data stream mining and discusses its techniques. Next, the IoT and IDS are discussed in Sections 2.3 and 2.4. After that, the classification and metaheuristics with the techniques that have been used in this work are presented in Sections 2.5 and 2.6. Section 2.7 states the silhouette index. In 2.8, some of the data pre-processing methods are discussed. In Sections 2.9 and 2.10, the performance matrix and the programming platform will be stated.

2.2 Data Stream Mining

As the name implies, data stream mining is linked to two fundamental topics of computer science: data mining and data streams. Data mining can be defined as an interdisciplinary subject of computer science whose major goal is developing tools and methods for analyzing massive datasets for knowledge. It is closely linked to statistics, pattern recognition, and machine learning, and it employs a variety of techniques. The vast majority of machine learning approaches are intended to be used with static datasets. However, because of their unique nature, no direct application can be performed onto the data streams. Data streams are sequences of data pieces that are ordered [1][2][22].

The inapplicability of typical data mining algorithms is determined by three primary characteristics of data streams.

1. A vast, perhaps infinite number of data elements.
2. High rates of system data arrival.

3. Potential variations of different kinds in data distributing throughout data stream processing are all factors to consider in this subject, also known as concept drift.

An algorithm can process a data stream in one of two ways. The first type of algorithm is an online algorithm that performs computations and updates the output when all data elements are read from the stream. The data stream is partitioned into blocks (chunks) via the second type of method [22].

2.2.1 Data Stream Mining Technique

The unlimited flow of data streams, as well as the rapid sequence of instances, make the procedure rather difficult. To successfully evaluate, integrate, collect, and convert data in real time in only a single scan while maintaining process continuity, data stream mining techniques are necessary. The next parts present an overview of several real-time clustering and classification data stream mining techniques [23].

2.2.1.1 Stream Clustering Technique

Because of its usefulness, data stream clustering has got a lot of interest in research. The adoption of arbitrary clustering methods for data streams is rather complicated due to the single scan pass that is available. Using a sliding window, stream clustering could be done for the full stream or simply the most recent items in the stream. [23][24]. Figure (2.1) shows the different type of stream clustering technique.

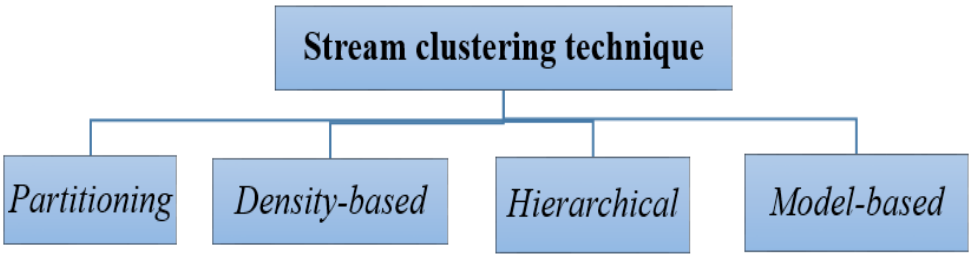

Figure 2.1: Stream clustering technique

A- Partitioning Method

Partitioning clustering methods form a common clustering approach. The idea is to group items together based on their similarities. The user must specify the number of clusters to be formed. A cluster is represented by each partition. These clusters must have particular characteristics, such as containing at least one data item and categorizing each data item into exactly one cluster. K-means, K-medoids, and CluStream are just a few of the clustering techniques that use partition-based clustering algorithms. Clusters are generated by all of these algorithms. A cluster is a group of points near the cluster's center in k-means clustering. The k-medoids algorithm is similar to k-means, except that k-means calculates the average distance, whereas the k-medoids algorithms determine the mediods of the cluster points. Aggarwal et al. proposed the CluStream algorithm, which clusters developing data streams using a k-means approach. [24]. This method have been used as the partitioning task for this thesis.

B- Density-based Method

Density-based methods work well for identifying arbitrary forms of clusters and dealing with data noise. These methods do not need to set a number of clusters but do not function well enough with multi-dimensional data. As a result, clusters are defined as areas with a large number of documented objects, divided by areas with little or no data. DBSCAN, PTICA, and PreDeCon are a few examples of these techniques [23][24][25].

C- Hierarchical Method

This clustering method uses an agglomerative or divisive strategy for creating a hierarchical tree-like cluster structures. For time series data streams, ODAC is an online divisive-agglomerative clustering technique. Two other examples are E-Stream and HUE-Stream [23][24].

D- Model-based Method

Model-based clustering can be described as a hypothesized model providing a statistical method for calculating the cluster number and the best fits of data points while accounting for misfits [23][24].

2.2.1.2 Stream Classification Technique

Classification algorithms are extensively utilized and studied in predictive data mining and knowledge discovery. Offline and online streams can both benefit from data stream classification. Online stream classification analyses and modifies data when it arrives individually, according to their characterizations [23][26]. Figure (2.2) shows the different type of stream classification techniques.

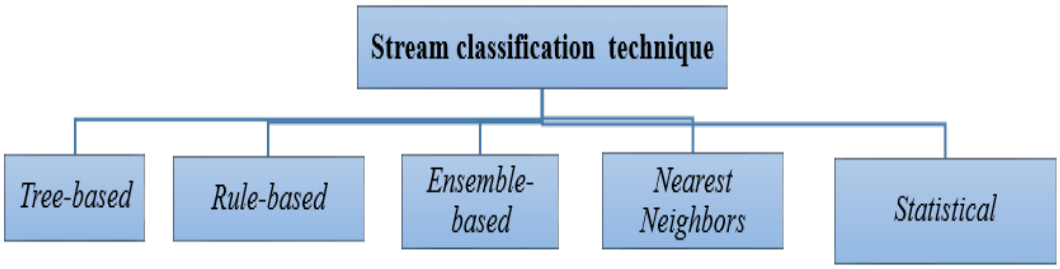

Figure 2.2: Stream classification technique

A- Tree-based

VFDT (Very Fast Decision Tree) and CVFDT (Concept-Adapting Very Fast Decision Tree) are techniques to classify data streams in light of the Hoeffding tree paradigm. After training a few samples, a usable model emerges. To regulate attribute splitting and preserve the best results for splitting less than the threshold, the user needs specify a threshold value. To delete less valuable leaves and keep counts for all leave nodes in memory, a statistical analytical comparison is performed. Pruning strategies are used to avoid higher memory use. In CVFDTs, a sliding window applied for keeping the model consistent and tackling the problem of concept drift that arises in VFDTs [26][27].

B- Rule-based

Clustream's on-demand classification method uses micro-clusters. Depending on each micro-cluster, the summary statistics are labeled on

classes, and they are updated when new data becomes available. To overcome the problem of concept drift in data evolution, this technique uses micro-cluster categorization after specifying a time horizon. SimC is built on instance-based learning techniques, which save a representative subset of data [26].

C- Ensemble-based

SAE2 is a social dynamic ensembles classifier. It connects classifiers that make similar predictions to form a network out of the whole ensemble of classifiers. The adaptation method in SAE2 is more scalable.

SFNClassifier is a dynamic scale-free network ensemble-based classifier. To extract centrality metrics for weighted majority voting, the ensemble is represented as a network.

KME is a new classifier that detects idea drift and recognizes recurrent concepts by combining supervised and unsupervised knowledge. To improve recognition ability, it assesses ensemble member weights and reuses conserved labeled instances from previous blocks [26][28].

D- Nearest Neighbors

The Nearest Neighbors (ANN) classification methodology is based on a general structure for arbitrary nearest neighbor methods with distance measures that can be halted at any time. It produces the best results in the shortest time possible. They arrange the index of training set for ANN classifiers using generic and particular ordering heuristics strategies. Even though the process duration is quite short, the algorithm cannot be interrupted during the step-up phase [23].

E- Statistical

SAL employs a Bayesian model that enables multi-class classification without the usage of a predetermined class labels. For predicting the marginal and conditional distributions, it utilizes all labeled and unlabeled data [23].

In this thesis, the proposed system classifies the dataset using the traditional classification technique as Artificial neural network and SVM.

2.3 Internet of Things (IoT)

The Internet of Things (IoT) is a term used to describe the millions of physical gadgets linked to the internet that gather and exchange data worldwide. During the late decades of the 19th century, the concept emerged on the addition of intelligence and sensors to objects. However, it progressed rather slowly due to the technological delays in development. IoT was invented by Kevin Ashton in 1999, yet technology could only catch up after an additional ten years. Among the great challenges found in IoT is its security. Since the sensors gather data of incredible sensitivity (such as what is spoken and done in and around the house), securing this data is considered to be critical for consumer confidence. Yet, the security track records in IoT remains rather dismal so far, as most devices do not pay attention to the security basics such as the encryption of data when transited or at rest. One or more sensors will almost certainly be included in an IoT device, which is used to collect data. What such sensors acquire will vary depending on the gadget and its purpose. Industrial machinery sensors may detect temperatures or pressure; security cameras may include proximity sensors in addition to sounds and videos; even the home weather stations almost certainly include humidity sensors. All of these sensors for data, as well as a great deal more, need to be delivered somewhere. This implies that IoT

devices need to send data across Wi-Fi, 4G, 5G, and other networks [29]. Figure (2.3) shows the IoT key stages [30].

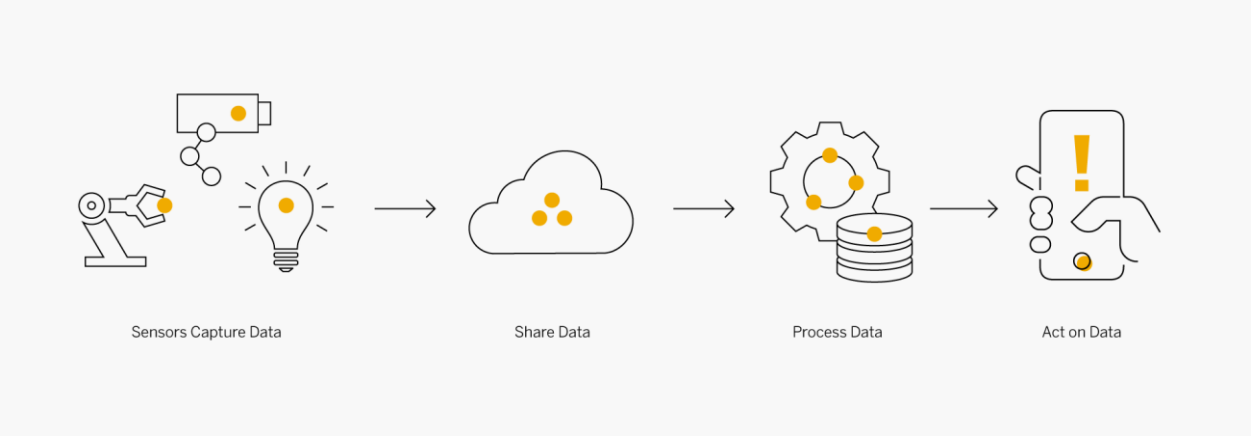

Figure 2.3: IoT key stages [30].

2.4 Intrusion Detection Device (IDS)

Intrusion Detection Systems (IDS) are devices (application) that watch for malicious activities or policy breaches on networks. Security information and event managing systems are often adopted for reporting or gathering any form of malicious activities or violations. Some devices can respond to intrusions at the moment of discovery. IDS come in a variety of forms, ranging from anti-virus software to tiered monitoring systems for tracking network traffics [31]. Some of the most prevalent classifications include:

- Network Intrusion Detection Systems (NIDS): They are systems for analyzing incoming network traffics.
- Host-based Intrusion Detection Systems (HIDS): They are systems that detect intrusions and monitor critical operating system files.

Detecting intrusions takes place through two approaches: according to signatures or anomalies. The first type of systems compares provided network traffic and log data to known attack patterns to detect potential threats. Sequences (thus the name) are a type of pattern that can comprise byte sequences, also called harmful instruction sequences. This type of detection is used for reliably detecting and identifying known assaults [31]. Figure (2.4) shows the IDS [32].

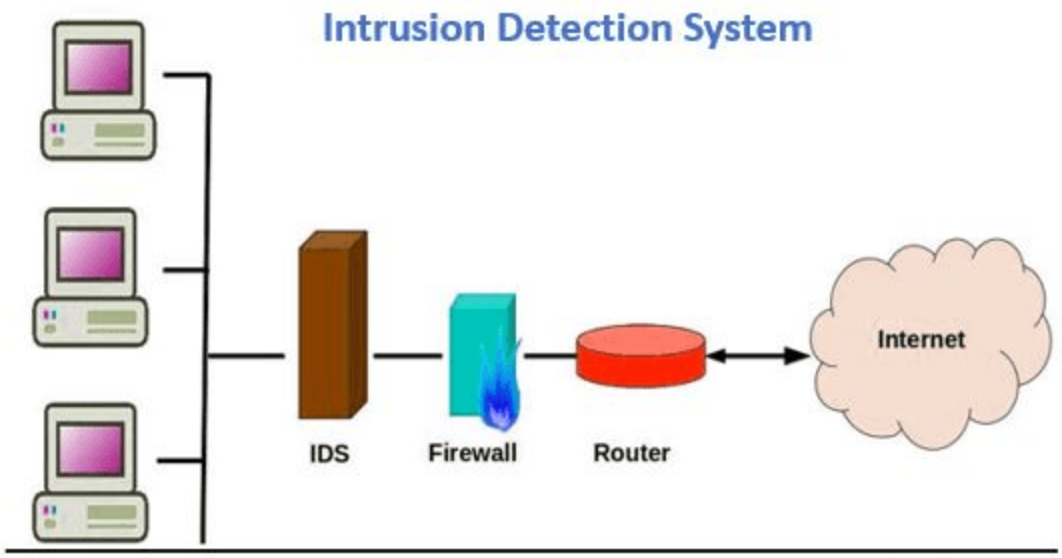

Figure 2.4: IDS (intrusion detection system)

2.5 Preprocessing Data

Data pre-processing is a stage in Machine Learning processes whereby changes or encodings are done to the data for easing the process of machine parsing. This means that the algorithm cis able to quickly interpret the data's characteristics. A dataset is a grouping of data objects [33]. Figure (2.5) shows the statistical data types.

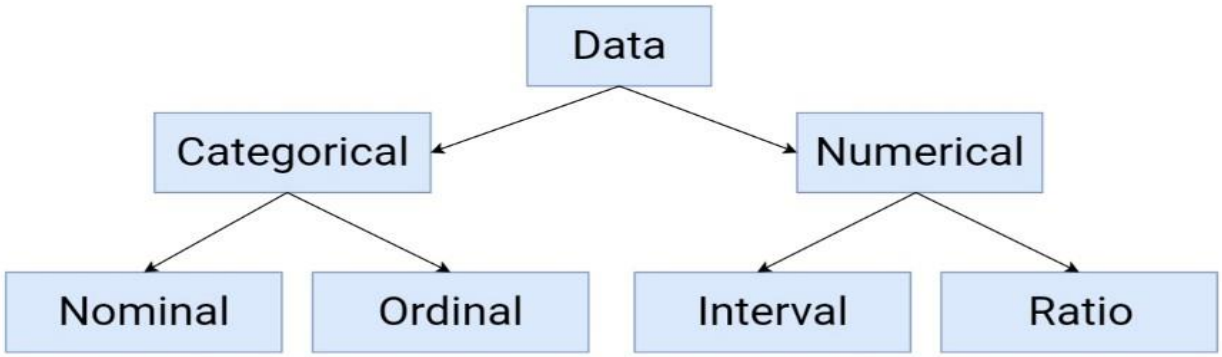

Figure 2.5: Statistical Data Types

Given the type of data set used, only a few steps may be required. In general, they are as follows:

1. Data Quality Assessment: The expectation that the data will be perfect is not reliable. Missing values, inconsistent values and Duplicate values are examples.
2. Feature Aggregation: Feature Aggregations are used to take the aggregated numbers and put them into context with the rest of the data.
3. Sampling: Sampling is a typical technique for picking a portion of a dataset for analysis.
4. Dimensionality Reduction: This process minimizes the amount of features. The Curse of Dimensionality refers to the fact that as the dimensionality of the data grows, data analysis activities become progressively more difficult.
5. Feature Encoding: Feature encoding is the process of transforming data so that it can be used as input for machine learning [33][34] [35].

In this thesis the utilized dataset is clean and already pre-processed. For that reason, no pre-processing methods have been applied to the dataset.

2.6 Cross validation

Cross-validation is a resampling technique for evaluating machine learning models on a small sample of data. The process includes only one parameter, k, which specifies the number of groups into which a given data sample should be divided. As a result, the process is frequently referred to as k-fold cross-validation. When a precise value for k is specified, it can be substituted for k in the model's reference, for example, k=10 for 10-fold cross-validation. Cross-validation is a technique used in applied machine learning to estimate a machine learning model's skill on unknown data. That is, to use a small sample to assess how the model will perform in general when used to generate predictions on data that was not utilized during the model's training. It's a popular strategy since it's straightforward to grasp and produces a less biased or optimistic estimate of model competence than other approaches, such as a simple train/test split [36].

2.7 Classification Technique for Intrusion Data

The common thought for classification is to forecast the objective class for the given dataset by analyzing the training dataset [3]. The classifier in data stream classification is trained on classified data that comes through time line.

The classifier is used to forecast the label of the class for the unseen data that comes in the next period. The input data is also provided to the objectives in classification, which is a sort of supervised learning. Credit approval, medical diagnoses, and target marketing are just a few of the sectors where classification is useful [37][38][39].

After chunking and gathering the dataset by the PSO algorithm, the chunks are classified using multiple classifiers, each of which presents a result. Then, the

results of each classifier are evaluated and the optimal one with the highest accuracy is chosen. In this thesis, the following classification algorithms are used.

2.7.1 Artificial Neural Network

Neural networks are complicated models which attempt to stimulate the brain of the human to develop the classification rules. It contains many characteristic neurons layers, each layer receives inputs and delivers outputs to the next layer, depending on the weights given to this specified link. It relies on the cost function and the optimizer. The neural networks are repeated many times depending on the number of iterations (which is foreordained), called epochs [40]. The cost function is analyzed after each epoch to consider where the model might be moved forward. At that point, the optimizing function modifies the inside mechanics of the network, such as the weights and biases to reduce the cost function. In light of how complex the functions ae mapped through the model, the latter may involve many hidden layers. The modeling of complicated relationships, as with deep neural networks, is of more simplicity when there are more hidden layers [41].

Artificial Neural Networks are weighted directed graphs with nodes representing artificial neurons and directed edges. The weights represent connections between neuron outputs and neuron inputs. It receives the data from the outside environment in the form of a pattern and a vector image. Each input is multiplied by the weights assigned to it. The information that the neural network uses to solve a problem is called weights. The strength of the connections between neurons within the Neural Network is typically represented by weight. Inside the computing unit, all of the weighted inputs are summed together (artificial neuron). Any numerical number between 0 and infinity corresponds to the sum. Figure (2.6) shows the main steps of using neural network classifier. The threshold value is set to limit the response so that it reaches the desired value. To get the desired result,

the activation function is set to the transfer function. There are two types of activation functions: linear and non-linear [42].

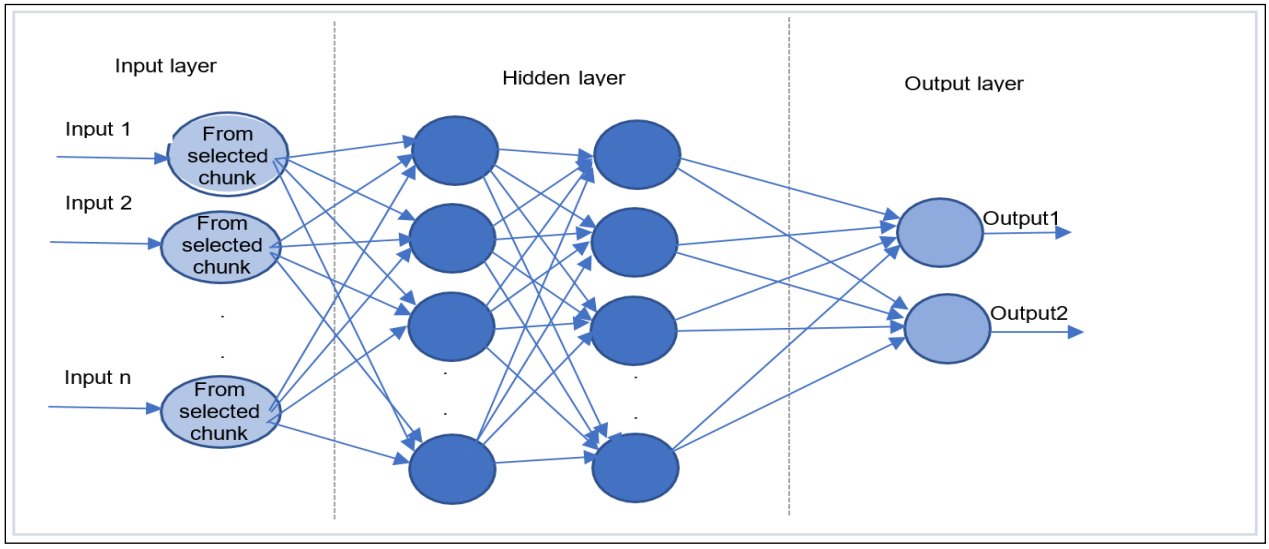

Figure 2.6: Main steps of using neural network classifier with the input selected chunks data

2.7.2 Support Vector Machine

The perceptron algorithm can be thought of as an extension of this technique. The goal of SVM optimization is to find a decision line that divides the classes by increasing the margin between it and the sample points nearest to the hyperplane. Support vectors are the names given to these points. The SVM model tries to discover or find the hyperplane (decision boundaries) between two or more classes to the problem with N-dimensions spaces, whereby N is the number of features. To decrease the impact of the data on the side of the discriminant margins are weighted down ("soft margin"). When the data has N-dimensions space, this make the process of finding hyperplane more difficult. For that reason, the "kernel function" is used [43][44]. Formula (2.1) shows the main loss function for SVM.

$$\min_w \lambda \|w\|^2 + \sum_{i=1}^{n} (1 - yi\langle xi, w\rangle)_+ \ldots (2.1)$$

Where x is the vector of the features, w is the normal direction of the plane, $y \in \{1, -1\}$. Since it only uses a sub-set of training points in its decision function, this technique is particularly beneficial in high-dimensional spaces and is memory economical [45]. Figure (2.7) and figure (2.8) illustrate the process of finding the hyperplane when the data set has two features (X, Y) and two colors (blue, green) [46].

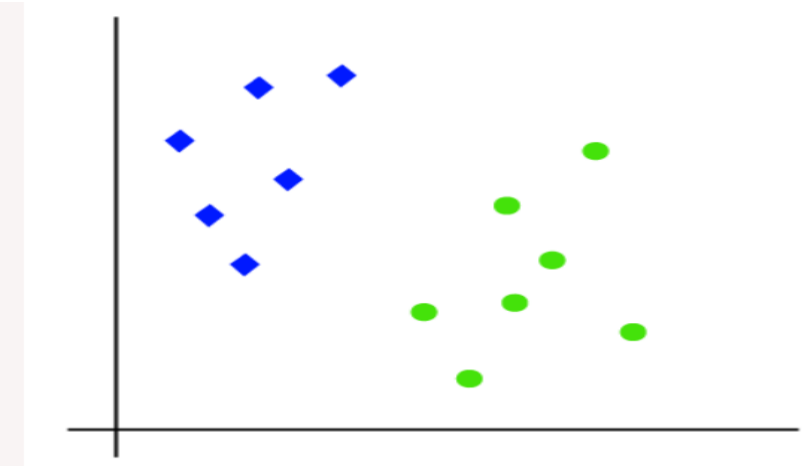

Figure 2.7: Dataset has 2 features (X, Y) and two color (blue, green) [45]

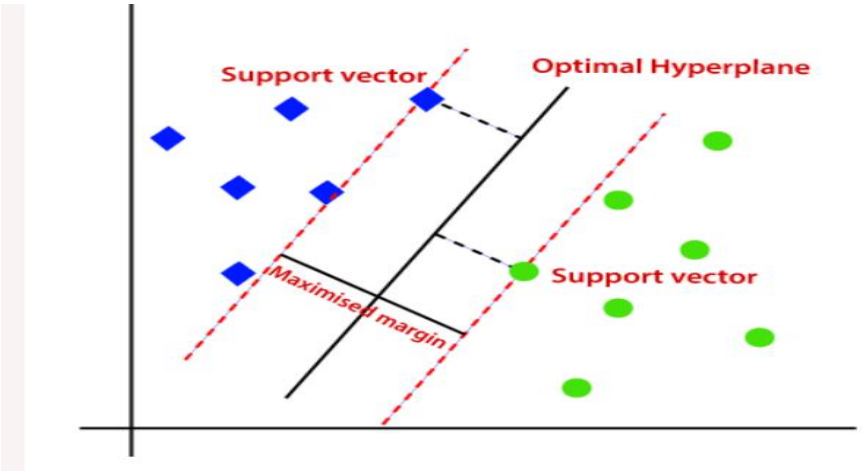

Figure 2.8: Main steps of using SVM classifier [45]

2.7.3 Naïve Bayes

Naive Bayes is a simple classification technique that predicts the classification of incoming data based on historical data. This model is a probabilistic model extended from the Bayes theorem. Given the aforementioned assumption, the classifying process is performed through calculating the maximal posterior, which is the maximum P(Ci|X). Through the count of class distributions, this assumption cuts the computational costs in a drastic manner [41][45].

Despite the fact that the assumption is false in most circumstances since the qualities are interdependent, Naive Bayes has managed to outperform the competition. For estimating the demanded parameters, only a limited amount of training data is needed. In comparison with more advanced algorithms, the speed of Naive Bayes classifiers is remarkably high [47] [48]. Its equation is shown in formula (2.2), whereby one could obtain how much A would happen (its probability) while B has occurred.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \dots (2.2)$$

Where P(A) is the prior probability of class. P(B|A) is the likelihood which is the probability of predictor given class. P(B) is the prior probability of predictor.

2.8 Metaheuristic Technique

Metaheuristic research has been widely published in literature, which encompasses methods, applications, comparisons, and analysis, due to successful implementations and high intensity. Metaheuristic techniques are generic search approaches utilized to solve the intractable optimizing problem which has a large size, to increase the outcome with restricted resources. Such problems include cases

where there are multiple solutions available, and the methods obtain the best or optimal solution through a relatively shorter period of time [49].

Metaheuristics are used to address optimization problems by searching for the best possible solutions to a specific problem. Multiple agents can be used in the search process, which effectively construct a system of evolving solutions utilizing a set of rules or mathematical equations across a number of iterations [50].

Metaheuristic methods can be divided into two types: single-solution metaheuristics (local search) and population-based metaheuristics (random search). ACO, GA, and PSO are examples of population-based metaheuristic methods [51].

1. Ant Colony Optimization: is a term used to describe the process of optimizing a Ant colony optimization (ACO) is a concept inspired on ant colony behavior. The utilization of chemical compounds created by ants is used to communicate between ants or ants and their surroundings. Pheromone is the chemical molecule in question.
2. Genetic Algorithm: Algorithm genetics. The genetic algorithm (GA) is a nonlinear and random search technique based on natural selection principles. John Holland discovered this approach in the 1960s. The GA proofs begin with coding or parameter coding. The chromosomal generation is then completed. A group of genetics known as a chromosomal generation.
3. Particle Swarm Optimization: In 1955, Dr. Eberhart and Dr. Kennedy created the Particle Swarm Optimization method. The behavior of a flock of birds or a swarm of fish was inspired by PSO. The PSO algorithm is a population-based randomized optimization algorithm. If the method has reached convergence, the optimal solution is obtained in PSO. The position and velocity of the particles impact whether the algorithm has reached

convergence. As for the work presented in this thesis, the PSO (Particle Swarm Optimization) algorithm is used for the optimization task.

2.8.1 Particle Swarm Optimization

Kennedy and Eberhard proposed a particle PSO algorithm based on the concept of swarm intelligence. It is easy to understand and to deal with because it has fewer parameter to adjust. To understand the PSO concept, the particle functions as a flock of birds that searches the whole problem search space for finding the best position that leads to the goal [52]. PSO is an iterative metaheuristic technique for solving global optimization issues.

It falls under the category of bio-inspired approaches, which aim to replicate some natural behavioral paradigms in groups of people. PSO seeks to emulate the reasoning behind a swarm foraging for food in an iterative manner. Each swarm member is referred to as a particle [53].

The basic parameters that are used in this algorithm are: the size of the problem, number of particles, acceleration coefficients (c1) and (c2), inertia weight (w), number of iterations (t), and the random values for measuring the contributing extent for the cognitive and social components (r1) and (r2). The mathematical formula of the PSO is shown in equations (2.3) and (2.4).

- $vp_i^{k+1} = wvp_i^k + c_1 r_1 \left[Pbest^k - x_i^k\right] + c_2 r_2 \left[Gbest - x_i^k\right]$ ... (2.3)
- $xp_i^{k+1} = xp_i^k + vp_i^{k+1}$... (2.4)

Where:

- vp: Is the velocity vector of the swarm $vp_i^t = (vp_{i1}, vp_{i2}, vp_{i3}, \dots, vp_{in})^T$ at iteration t for each particle in the swarm i .

- xp: Is the position vector $xp_i^t = (xp_{i1}, xp_{i2}, xp_{i3}, \dots, xp_{in})^T$. These equations are used to update the velocity and position.
- w: Inertia weight.
- c1,c2 ,r1 ,r2: Are the acceleration coefficients where r1 and r2 are a random numbers.
- $Pbest$: Best personal position, as the term extends, the particle is drawn to its most advantageous place.
- $Gbest$: Best global position, attracts particles to the best place until it is discovered at some t iteration.

The PSO behavior is modeled by the method used by bird swarms in their search for the best food sources. The direction of the bird movement throughout the search process is influenced by its present movement towards locating the best food sources. The latter is influenced by the bird's knowledge, swarm knowledge, and inertia. The PS algorithm simulates this method by representing personal and global optimum positions, as well as inertia for every particle representing a solution. All particles have their own positions, velocities, and objectives, aiming towards reserving the global-best value by obtaining the optimal objective value and global position [54].

2.9 Silhouette Index

Silhouette analysis is a technique for interpreting and validating consistency within data clusters. In comparison with alternative clusters, the silhouette value measures how close objects are to their clusters (cohesion) (separation).

It can be used to figure out how far apart the generated clusters are. The silhouette plot indicates the similarity between points in one cluster and points in

the closest neighbors, thereby enabling the visual examination of parameters like cluster counts. The calculation of the silhouette index is performed through the comparison of the average distances among data points in one cluster, and the average distances among data points in other clusters [55], as shown in Figure (2.9) [56].

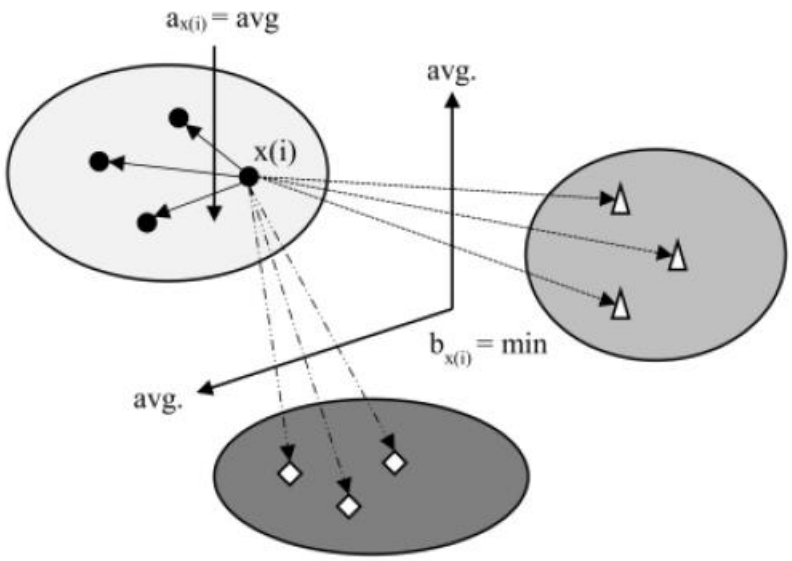

Figure 2.9: Concept of the silhouette index

K can be defined as the cluster that consisting of data $x(i)$

$$S_{x(i)} = \frac{b_{x(i)} - a_{x(i)}}{\max(b_{x(i)} - a_{x(i)})} \dots (2.5)$$

Where:

$x(i)$ = the point in K, and i = 1, 2, 3...m, $b_{x(i)}$ = (i) 's average distance from each data point in the same cluster, and $a_{x(i)}$ = the shortest average distance between $x(i)$ and each data point in the other clusters.

$$S_k = \frac{1}{n} \sum_{i=1}^{n} S_{x(i)} \ldots (2.6)$$

$$S_{avg} = \frac{1}{m} \sum_{k=1}^{m} S_k \ldots (2.7)$$

Where: K = clusters number, n = the number of cluster points and m = all clusters number [56].

After the evaluation task, the test data must choose one of the chunks for training the model. This is done by using a mathematical equation based on the Mahalanobis distance, as shown in formula (2.8).

$$eq(i) = 1 - (D_M(i) / \sum_{i=0}^{n} D_M(n) \ldots (2.8)$$

Where:

- i: represents the current chunk

- n: the number of the chunks

-$D_M(i)$: the mahalanobis distance for each chunk using the equation in (2.9):

$$D_M(\vec{X}) = \sqrt{(\vec{X} - \vec{\mu})^T S^{-1} \vec{X} - \vec{\mu}) \ldots} \ (2.9)$$

Where $\vec{X}$ the vector of the observation is, $\vec{\mu}$ is the vector of mean values, and S^{-1} is the inverse covariance matrix.

2.10 Performance Metrics

Building machine learning models is based on the principle of constructive feedback. The process involves creating a model, gathering data from measurements, making adjustments. This procedure is repeated until the desired

accuracy is attained. The performance of a model can be explained using evaluation measures. The ability of evaluation metrics to discern between model results is a key feature. In the machine learning, the model could be evaluated using the Confusion matrix [57].

The confusion matrix is N X N in which N is the number of expected classes. Given that N=2 for the problem at hand, thus the matrix will be 2 X 2. The examples of predicted classes are indicates by the rows of the confusion matrix, meanwhile the occurrences in actual classes are indicated by the columns. The Confusion Matrix is technically not a performance statistic, but it serves as a foundation for other metrics to analyze the outcomes [58][59].

There are four key terms to remember:

- True Positives: When the predicted value is YES and the actual result is also YES.
- True Negatives: These are the occasions where the predicted value is NO and the resulting value is also NO.
- False Positives: These are the occasions when the predicted value is YES but the actual result is NO.
- False Negatives: These are occasions where the expected value is NO but the obtained value is YES, as shown in Figure (2.10).

1. **Accuracy**: It is the number of correctly classified data instances divided by the total number of data instances:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots (2.10)$$

2. **Precision**: is the anticipated proportion of true positives to total positives:

$$P_i = \frac{TP_i}{TP_i + FP_i} \dots (2.11)$$

3. **Recall**: is the proportion of true positives in ground truth to all other positives:

$$R_i = \frac{TP_i}{TP_i + FN_i} \dots (2.12)$$

4. **F1-measure**: It can be described as the Harmonic Mean of precision and recall, whose score ranges within [0, 1]. It gives an indication of the exactness and robustness of the classifier, based on the number of successful classifications without missing any significant instance numbers. The high F1 Score indicates a better model performance. Form a mathematical point of view [58][59], it can be defined in the following way:

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \dots (2.13)$$

TP	FP
FN	TN

Figure 2.10: Confusion Matrix

2.11 Summary

This chapter covered all the concepts that the thesis includes. Starting with an overview of stream mining with data stream, it is followed by discussing the properties and the problems of its techniques. Next, the IoT and IDS devices are explained in addition to the classification techniques and the algorithms that have been used. After that, the metaheuristic technique was stated along with the PSO algorithm used in this thesis. Finally, each of the silhouette index, data pre-processing and the performance matrix are stated in the last sections

Chapter Three

THE PROPOSED SYSTEM AND METHODOLOGY

3.1 Introduction

This chapter contains the practical section of the work, which is split into two phases: optimization and classification. Before the optimization phase, the general steps of the proposed work and the proposed system are explained. Next, the dataset that has been used in this thesis is described. The optimization phase involves the use of the PSO algorithm to chunk the dataset into two optimally evaluated chunks. These two chunks are evaluated by using the silhouette index. After that, the appropriate chunk is chosen based on a mathematical equation to train the three models. Finally in the classification phase, the data mining classifiers classify the test data based on one of the two chunks, as shown with the evaluation matrix.

3.2 General steps of the proposed work

The main proposed system of this work is presented in Figure 3.1. The selected sensor data is split into training and testing to for each (1) second to split into chunk (1) and chunk (2) based on the PSO and silhouette index evaluation. These two chunks are trained with the classifier based on the mathematical equation which chooses one of them. The evaluation metrics are calculated to select the higher classifier accuracy. Figure (3.1) illustrates the general steps of the proposed system. Figure (3.2) illustrates the system implementation phases and the generally followed steps to achieve an optimal and higher performance of the chosen models.

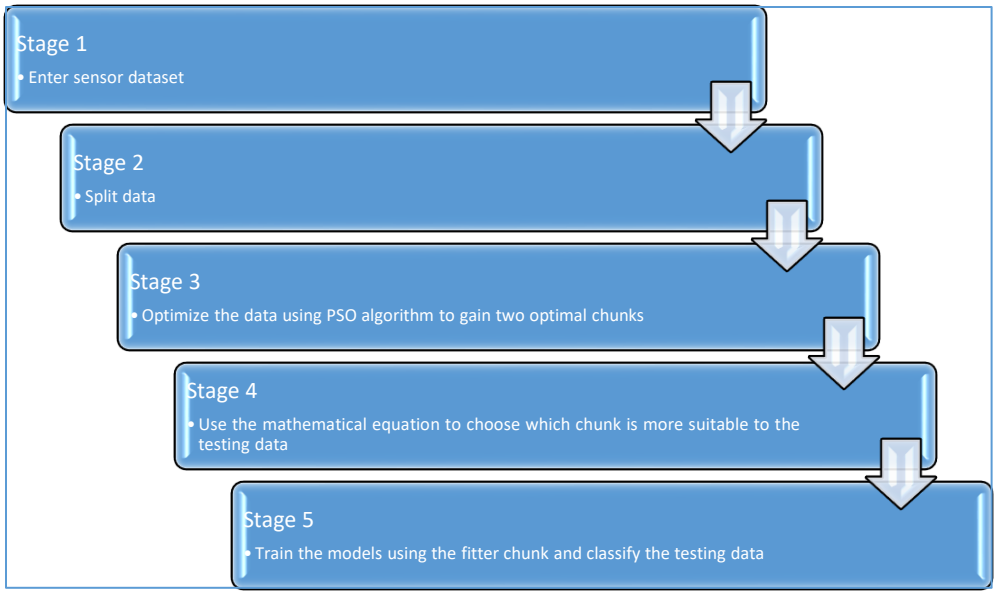

Figure 3.1: Main stages of the proposed system

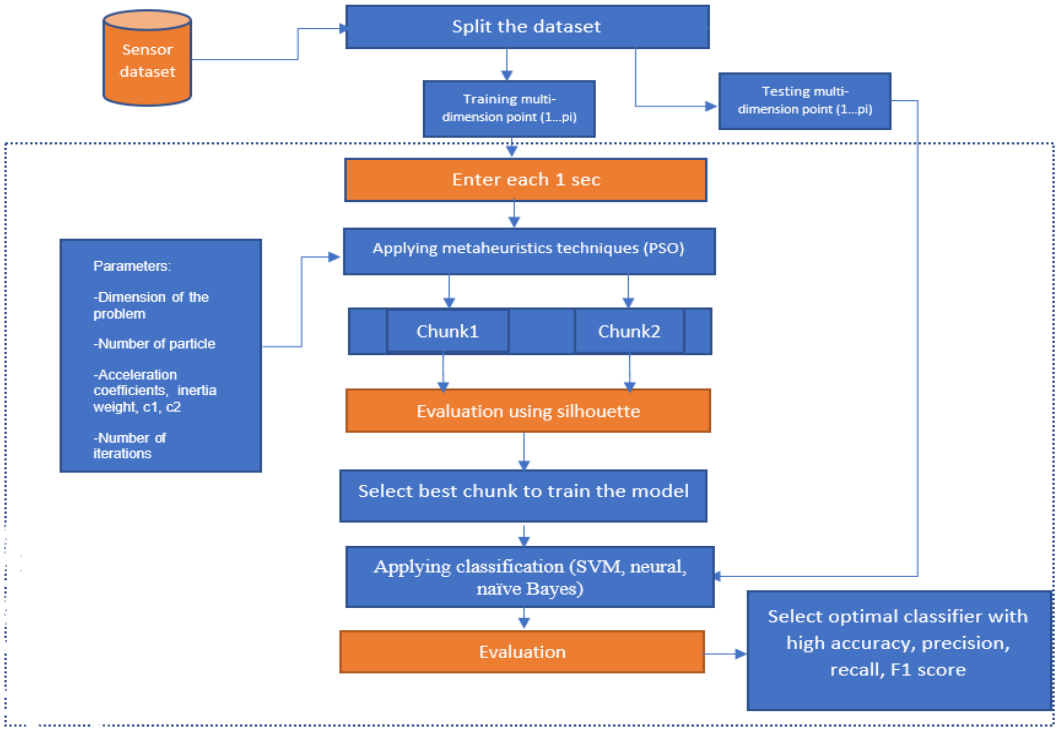

Figure 3.2: Proposed system of stream mining for IDS data

3.3 The proposed methodology

The proposed system in this thesis is used to predict the sensor stream data with high performance. The sensor stream data is represented with the IoT device logs dataset. The methodology that has been used here is the metaheuristics technique (which represented by the PSO algorithm) to partition the dataset into two chunks after splitting the data randomly into 70% training and 30% testing. After chunking the data, the data mining classifiers classify the testing data using three models (ANN, SVM, and Naïve Bayes). These models are trained, each of which is based on one of the two chunks. This depends on the result of the mathematical equation which selects the best chunk that has the highest similarity to the testing data. Each of the three models gives a different result. Algorithm (3.1) show the main steps for the proposed system.

Algorithm (3.1): The main steps for the proposed system

1. **Input**: Training data Point (TP) with multi-dimensions, Testing data Point (P) with multi-dimensions
2. **Output**: Classified data using neural, SVM, naïve Bayes classifiers
3. **Begin**:
4. **Split** the data into TP and P
5. for each 1 sec from the TP:
6. **Apply** PSO algorithm (3.2) with a threshold for partitioning the data into chunk1 and chunk2
7. **Evaluate** the chunks using silhouette
8. **End** for
9. **Apply** a mathematical equation in eq (13) and (14) to select the nearest chunk to the testing points

10. **Train** the model using the selected chunk
11. **Classify** the testing points Pi using neural, SVM, naïve Bayes classifiers
12. Evaluate the three models using accuracy, precision, recall, and f1 score
13. **Select** the optimal classifier for evaluation
14. End

3.4 Dataset description

The dataset is pre-processed and already clean. For that reason, there is no need for the preprocessing method. This is a dataset for network based intrusion detection system (IDS) in IoT Devices which has been discussed in Section 2.3 and 2.4. This data contains multiple features for detecting five types of intrusion. It has six classes, in the following form:

0. Normal
1. wrong setup
2. DDoS
3. Data type palpating (utilizing ultrasonic sensor thus in palpation of the data type, mostly string values sent to the server)
4. Scan attack
5. Man-in-the-middle.

The detection process in this dataset depends on 14 features. Thirteen of them are described features like the frame time, length, Ethernet and IP sources and destination, whereas the last one is the normality class. It has 6 values (from 0 to 5) whereby each value represents one of the six types, as described above. Table (3.3) shows a screenshot for the dataset.

Table 3.1: Screenshot for the IoT device logs dataset

frame.tim	frame.len	eth.src	eth.dst	ip.src	ip.dst	ip.proto	ip.len	tcp.len	tcp.srcpor	tcp.dstpor	Value	normality
1.24E+14	54	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	40	0	49279	80	-99	0
1.24E+14	62	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	48	0	56521	80	-99	0
1.24E+14	62	1.67E+14	8.8E+13	1.92E+09	1.92E+08	6	48	0	80	56521	-99	0
1.24E+14	54	1.67E+14	8.8E+13	1.92E+09	1.92E+08	6	40	0	80	49279	-99	0
1.24E+14	54	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	40	0	56521	80	-99	0
1.24E+14	54	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	40	0	49279	80	-99	0
1.24E+14	269	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	255	215	56521	80	62	1
1.24E+14	54	1.67E+14	8.8E+13	1.92E+09	1.92E+08	6	40	0	80	56521	-99	0
1.24E+14	288	1.67E+14	8.8E+13	1.92E+09	1.92E+08	6	274	234	80	56521	-99	0
1.24E+14	54	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	40	0	56521	80	-99	0
1.24E+14	54	1.67E+14	8.8E+13	1.92E+09	1.92E+08	6	40	0	80	56521	-99	0
1.24E+14	62	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	48	0	56127	80	-99	0
1.24E+14	62	1.67E+14	8.8E+13	1.92E+09	1.92E+08	6	48	0	80	56127	-99	0
1.24E+14	54	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	40	0	56521	80	-99	0
1.24E+14	54	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	40	0	56127	80	-99	0
1.24E+14	269	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	255	215	56127	80	60	1
1.24E+14	54	1.67E+14	8.8E+13	1.92E+09	1.92E+08	6	40	0	80	56127	-99	0
1.24E+14	288	1.67E+14	8.8E+13	1.92E+09	1.92E+08	6	274	234	80	56127	-99	0
1.24E+14	54	8.8E+13	1.67E+14	1.92E+08	1.92E+09	6	40	0	56127	80	-99	0

This dataset obtained from IoT sensors devices logs is already preprocessed and clean. The network was monitored and network records were collected using an ultrasonic sensor with Arduino and NodeMCU. The data was sent to the server over Wi-Fi using a NodeMCU with an ESP8266 Wi-Fi module). Table (3.1) illustrate the features description of the dataset.

Table 3.2: Feature description

Attribute	Type	Description
Frame time	Continuous	The time of frame arriving
Frame len	Discrete	The length of the frame
Eth src	Continuous	Ethernet Source or other MAC address
Eth dst	Continuous	Ethernet Destination or other MAC address
IP src	Continuous	IP Source address
IP dst	Continuous	IP Destination address

IP len	Discrete	IP length
IP proto	Discrete	IP Protocol Number
TCP len	Discrete	Length of TCP payload
TCP srcport	Discrete	TCP Source Port Number
TCP dstport	Discrete	TCP Destination Port Number
Value	Discrete	The value that determines is there intrusion or not
Normality	Discrete	Based on the value the normality determines which class it is

3.5 PSO and data chunking

In this stage, the PSO algorithm has been used for optimizing the data. After splitting the data set randomly into 70% training and 30% testing, the training data is used as an input to the PSO algorithm. The PSO algorithm would divide the training data into two optimal chunks by utilizing a threshold (with a value of 0.2). Each of the training data values would have a random weight. Based on these weights, the data would either go to chunk (1) if its weight is bigger than a threshold, or to chunk (2) if it is smaller. After that, the silhouette index evaluates each chunk.

The value of the threshold was chosen by using the trial and error approach. This value is more appropriate than increasing or decreasing it. If the threshold is bigger than 0.2, most of the training data would go lost and never chosen. The same occurs when the value is set smaller than 0.2.

Algorithm (3.2) shows the main steps of Particle Swarm optimization (PSO) algorithm.

Algorithm (3.2): Particle swarm optimization

1. **INPUT:** Sensor Data as Point (P) with multi-dimensions
2. **OUTPUT:** optimum chunk from the dataset
3. **Begin:**
4. For each particle i = 1 to no. of particle do
5. Initialize the particle's position with a uniformly distributed random
6. Initialize the particle's best known position to its initial position: pi ← xi
7. if f(pi) < f(g) then
8. update the swarm's best known position: g ← pi
9. Initialize the particle's velocity
10. while a termination criterion is not met do:
11. for each particle i = 1 to no. of particle do
12. for each dimension d = 1 to no. of iteration do
13. Pick random numbers between (0,1)
14. Update the particle's velocity based on equation (2.3)
15. Update the particle's position based on equation (2.4)
16. if f(xi) < f(pi) then
17. Update the particle's best known position: pi ← xi
18. if f(pi) < f(g) then
19. Update the swarm's best known position: g ← pi
20. **End**
21. // fitness function
22. Divide the dataset based on threshold
23. Evaluate using silhouette index

Table 3.3: Particle swarm optimization parameters

Parameter of PSO algorithm	Value
particle number	20
Number of iteration	25
inertia weight	0.5
C1	1.25
C2	1.25
Threshold	0.2

The table above shows the basic parameters for the PSO algorithm that have been used and implemented in the Python platform. The choose of the parameters as (particle number, no. of iteration, w, c1, and c2) based on trial and error method, the increasing of the value for these parameters would make PSO slower but it would gave the same results.

The silhouette index evaluates the two chunks. After the two chunks are generated, the silhouette gives a result for interpreting and validating the consistency within chunks data. Algorithm (3.3) shows how the silhouette index is calculated.

Algorithm (3.3): silhouette index calculation

1. **INPUT**: two clustered data
2. **OUTPUT**: silhouette index score
3. **Begin:**
4. For data point $i \in C_i$ (i data point in the cluster C_i)

5. **Calculate** how well i matched to a cluster by $a_i = \frac{1}{|C_i|-1} \sum_{j \in C_i, i \neq j} dc(i,j)$ where $|C_i|$ is the number of points belonging to cluster i and $dc(i,j)$ is the distance between the cluster's data points i, j in the cluster C_i
6. **define** the average dissimilarity between point i and a cluster C_m where $C_i \neq C_m$
7. For data point $i \in C_i$
8. **Calculate** the neighboring cluster of i by $b_i = \min_{m \neq i} \frac{1}{|C_m|} \sum_{j \in C_m} dc(i,j)$
9. **Calculate** the silhouette value of data point i by
10. $sk(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, if\ |C_i| > 1$ and $sk(i) = 0\ if\ |C_i| = 1$
11. **End**

3.6 Optimal Chunk selection

When the two chunks are generated, the test dataset is used to evaluate the classifiers model. The test data must choose one of the chunks for training the model. This is done by using a mathematical equation based on the Mahalanobis distance, as shown in formula (2.8) and (2.9).

This step helps to select the chunk that is closer to the available test, so as to train the classifier on it based on the value of $eq(i)$ that represents the chunk with bigger value. The center values of the chunk i are taken, in addition to covariance Matrix with the values of the test.

3.7 Data mining classifiers

The phase of classification contains three classifiers which have been used in this thesis. Each classifier is trained on one of the two chunks based on equation formula (3.1). Then, the testing data would is classified in three different models, each of which gives different results.

3.7.1 Artificial neural network

One of the two chunks is trained using the model. The input is the testing data, and the output is the classified testing data. Algorithm (3.4) shows the main steps of the artificial neural network algorithm.

Algorithm (3.4): Artificial neural network

1. **INPUT**: Testing data Xt, weight w, and bias b
2. **OUTPUT**: a classified testing data Xt
3. **Begin:**
4. Assign weights to each input randomly
5. Calculate c = b +w * Xt
6. Calculate o = activation(c)
7. Update the weights using this two equation Δ wi = xi * o and wi(new) = wi(old) + Δwi and the output is back-propagated to minimize error
8. **End**

Table 3.4: Artificial neural network hyper-parameters

Hyper-parameters of neural network algorithm	value
No. of hidden layer	1000
Activation function	ReLU

Solver	Adam
Maximal number of Epochs	200
Learning rate	0.001

The table above shows the hyper parameters for the neural network algorithm that have been used and implemented in the Python platform.

- The No. of hidden layer parameter had to have large value (a range from 100 to 1000) to make the learning task and the classification accuracy increasing but as well as the value increased the task would be slower, for that reason it had to be in that range.
- RELU activation function is a less computationally intensive nonlinear activation function than the others.
- The Adam solver gave a better results than the others
- The Maximal number of Epochs and Learning rate took the default values

3.7.2 Support vector machine

One of the two chunks is training the model. The input is the testing data, and the output is the classified testing data. Algorithm (3.5) shows the main steps of support vector machine algorithm, whereby the kernel is linear because of its advantages for this work and more easily.

1. **Algorithm (3.5): support vector machine**
2. **INPUT:** Testing data Xt (x_i, y_i)., Weight w, and Bias b
3. **OUTPUT:** a classified testing data Xt

4. **Begin:**
5. if $y = w^T x_i + b = 0$ then (x_i, y_i) is support vectors then save the parameters $\boldsymbol{w}, \boldsymbol{b}$
6. elseif $y = w^T x_i + b = 1$ then save the parameters $\boldsymbol{w}, \boldsymbol{b}$
7. elseif $y = w^T x_i + b = -1$ then update the parameters $\boldsymbol{w}, \boldsymbol{b}$
8. **End**

3.7.3 Naïve Bayes

One of the two chunks is trained using the model. The input is the testing data, and the output is the classified testing data. Algorithm (3.6) shows the main steps of the Gaussian naïve Bayes algorithm.

Algorithm (3.6): Gaussian naïve Bayes

1. **INPUT:** Testing data Xt
2. **OUTPUT:** a classified testing data Xt
3. **Begin:**
4. Read all the Xt
5. **Calculate** the mean μ and standard deviation σ of the classes
6. **Repeat:** Calculate the probability for all classes
7. **Calculate** likelihood for each class
8. find the biggest likelihood
9. **End**

3.6 Evaluation matrix

All three models are assessed using the testing data at this stage. This stage evaluates the prediction error of each model using the accuracy, precision, recall, and f1 score variables which are described in Section (2.4). These measures depends on the confusion matrix, as shown in Table (3.4).

Table 3.4: Confusion matrix

	Yes	No
Yes	TP	FP
No	FN	TN

3.7 Summary

After all, this chapter proposed the main concepts of the thesis methodology. The general steps of the proposed work with the methodology of the thesis have been discussed clearly. Then, the preprocessing of the dataset has been described. Eventually, all proposed stages of this work, including (PSO optimization, optimal chunk selection, and data mining classification) with the performance matrix are presented and discussed.

Chapter Four

The Implementation results

4.1 Introduction

In this chapter, the results of this work are discussed as two results. After discussing all the PSO optimization results, the traditional method (classification without using optimization) results are stated for each model. The second one involves the thesis hyper technique (data mining classification within metaheuristic optimization PSO). After that, each model result is discussed, stating the reasons of the superiority for the model that gains the highest accuracy.

4.2 The Proposed System Implementation

The proposed system is based on two case studying with machine learning, as shown in Figure (4.1).

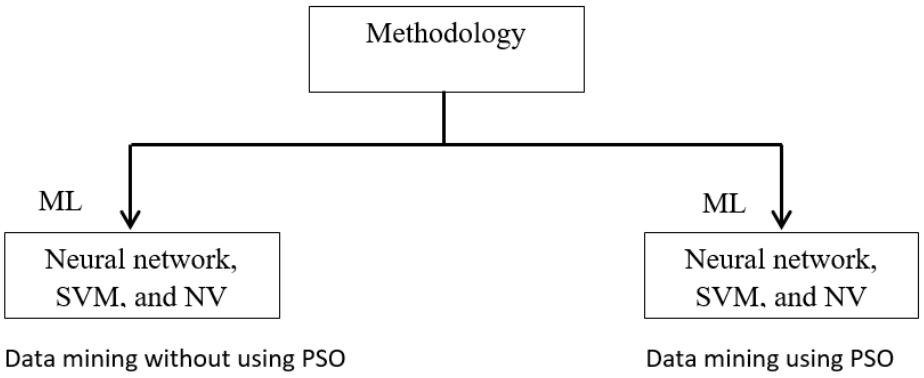

Figure 4.1: The used machine learning

The used system was built in an environment that conforms to the specifications listed in Table (4.1). Furthermore, the proposed system was implemented using the Python platform. The dataset used here is the IoT Devices logs.

Table 4.1: Environment specifications for the proposed system

Operating system	Windows 8.1, windows 10
CPU	Core (TM) I7-3630
RAM	8.00 GB
Implementation tool	Python, PyCharm Community Edition 2020.3.3 x64

4.3 Results of the PSO Optimization Phase

The PSO algorithm has been used to partition the dataset as mentioned in Chapter Three. This technique depend on random numbers that changed every iteration to find the best solution. In this work, the PSO algorithm partitions the data into two chunks using threshold and silhouette index. The silhouette index shows a different results when evaluating the two chunks.

Table 4.2: Number of records and attributed of IoT devices Logs Dataset

IoT devices Logs Dataset	
Number of attribute	14
Number of records	400,000

In this stage of optimizing the dataset, the dataset has a 400,000 filed out of which 280,000 have been used for training and the rest for testing.

The time complexity for the PSO algorithm was average (short time for a small dataset). Table (4.3) illustrate the execution time for PSO 25Th iteration.

Table 4.3: Execute time for PSO 25'Th iteration to treat a 1000'Th from the dataset

Iteration number	Execution time
1'it	2.626970672607 sec
2'it	2.626970672521 sec
3'it	2.626970672580 sec
4'it	2.626970672642 sec
5'it	2.626970672701 sec
6'it	2.626970673518 sec
...	...
25'it	1.593127227891 sec

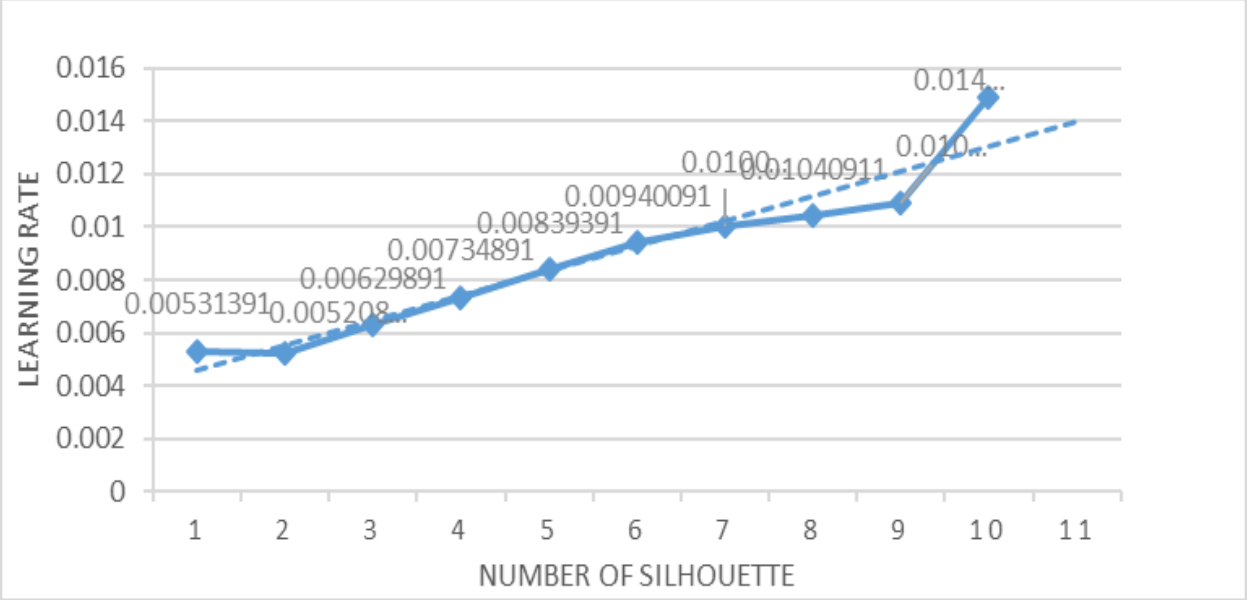

Figure 4.2: Silhouette Index Values to evaluate the PSO generated Chunks

Figure (4.2) illustrates the silhouette results for interpreting and validating consistency within the two chunks data. In this figure, the Silhouette Index Values

are presented to evaluate each of the two PSO generated Chunks. As shown above, it is obvious that the learning rate has an increasing trend, which means that they are well selected and the choice of the threshold is correct.

4.4 Results of the Traditional Classification Method

The traditional method means that the dataset has been classified by the three models without using any optimization technique (without using the PSO algorithm).

The three classifiers (neural network, SVM, naïve bays) gave different results. Some of them could be considered a high result, whereas the others get very low results.

4.4.1 Artificial Neural Network Classifier

The neural network algorithm is one of the data mining classification techniques which attempt to stimulate the brain of the human to develop the classification rules, as mentioned in Chapters Two and Three.

Table 4.4: Artificial neural network installation

Inputs	solver, activation, hidden_layer_sizes
Outputs	Classified testing data
Installation	Using python library from sklearn.neural_network import MLPClassifier

This algorithm has been utilized to classify the dataset by training the model using 70% of the dataset without any optimization technique and test the remaining 30% of the dataset. The result of this model in term of accuracy is 72.30.

4.4.2 Support Vector Machine Classifier

One of the data mining classification techniques is the support vector machine (SVM) methods which attempts to find the decision line that divides the classes by increasing the margin between it and the sample points nearest to the hyperplane, as mentioned in Chapters Two and Three.

Table 4.5: Support vector machine installation

Inputs	Kernel
Outputs	Classified testing data
Installation	Using python library from sklearn.svm import SVC

This algorithm has been utilized to classify the dataset by train the model using 70% of the dataset without any optimization technique and test the remaining 30% of the dataset. The result of this model in term of accuracy is 22.43.

4.4.3 Naïve Bayes Classifier

The naïve Bayes algorithm is one of the data mining classification techniques. This model is a probabilistic model based on the Bayes theorem. With the aforementioned assumption applied to Bayes theorem, the classification is done by calculating the maximum posterior, as mentioned in Chapters Two and Three.

Table 4.6: Naïve Bayes installation

Inputs	Model
Outputs	Classified testing data

Installation	Using python library from sklearn.naive_bayes import GaussianNB

This algorithm has been utilized to classify the dataset by train the model, using 70% of the dataset without any optimization technique and test the remaining 30% of the dataset. The result of this model in term of accuracy gives 40.08. Table (4.7) shows the accuracy results for the classifiers without using PSO.

Table 4.7: Performance evaluation percent for three classifiers without using PSO

Classifier	Accuracy	Precision avg	Recall avg	F1-score avg
Neural network	72.30	0.31	0.47	0.34
SVM	22.43	0.15	0.27	0.18
Naïve Bayes	40.08	0.45	0.67	0.44

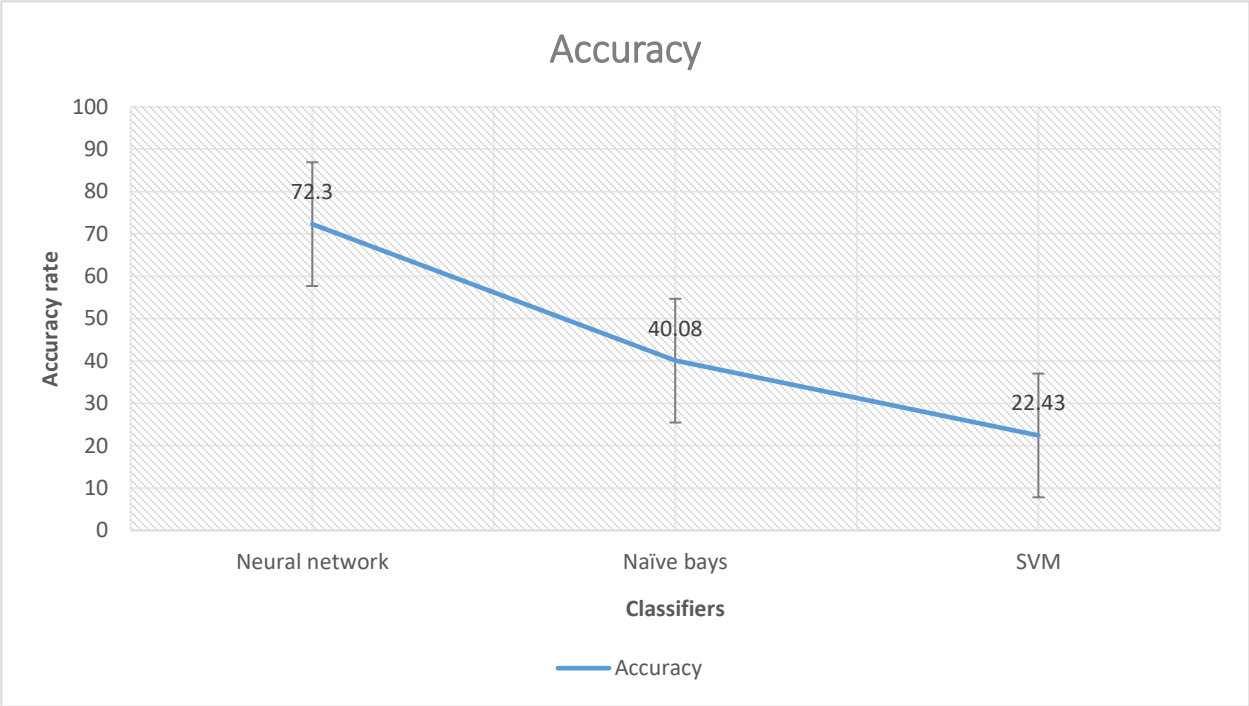

Figure 4.3: Accuracy rate for the three classifiers without using PSO algorithm

4.5 Results with Optimization Using PSO

The classification task with optimization means that the dataset has been classified by the three models using the optimization technique (through the PSO algorithm). The three classifiers (neural network, SVM, naïve Bayes) give different results, each of which have increased the accuracy rate.

4.5.1 Artificial Neural Network Classifier

The neural network algorithm is one of the data mining classification techniques, which attempt to stimulate the brain of the human to develop the classification rules, as mentioned in Chapters Two and Three. This algorithm has been utilized to classify the dataset by training the model using one of the two chunks that have been generated via the optimization technique (PSO). This is followed by

testing the remaining 30% of the data set separately (testing each 1000 data value), as shown in Table (4.6). The result of this model in terms of accuracy is 90.44.

Table 4.8: Accuracy results for each 1000 testing dataset using neural network classifier

Testing Dataset	Accuracy results
1000'1st	90.08
1000'2nd	88.90
1000'3d	86.78
1000'4th	89.35
1000'5th	89.50
1000'6th	87.94
1000'7th	90.30
….	….
Last 1000'th	90.44

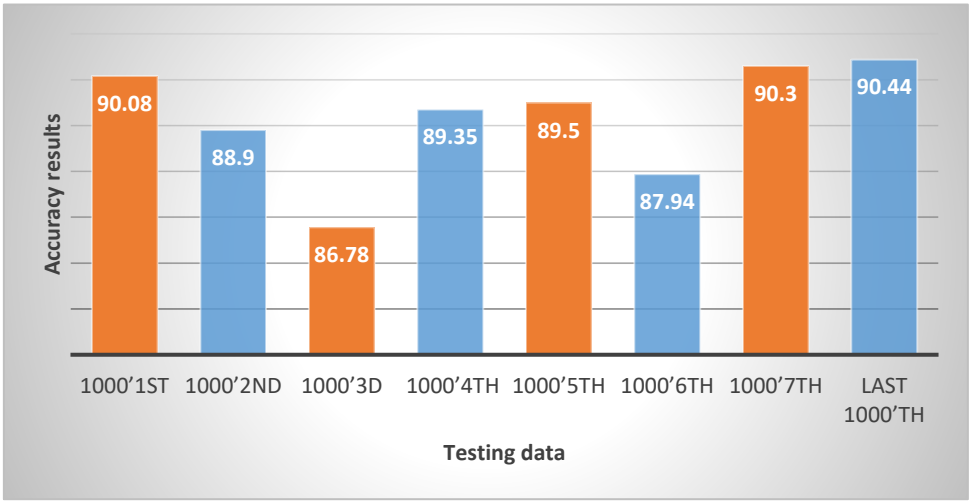

Figure 4.4: Accuracy results for each 1000 data from testing the dataset using neural network classifier

4.5.2 Support Vector Machine Classifier

One of the data mining classification techniques is the support vector machine (SVM) methods which attempts to find the decision line that divides the classes by increasing the margin between it and the sample points nearest to the hyperplane, as mentioned in Chapters Two and Three.

This algorithm has been utilized to classify the dataset by training the model using one of the two chunks that have been generated via the optimization technique (PSO), followed by testing the remaining 30% of the dataset separately (testing each 1000 data value). The results of this model in terms of accuracy is 58.70.

Table 4.9: Accuracy results for each 1000 testing dataset using SVM classifier

Testing Dataset	Accuracy results
1000'1st	57.10
1000'2nd	57.78
1000'3d	55.23
1000'4th	56.64
1000'5th	57.40
1000'6th	56.20
1000'7th	55.90
….	….
Last 1000'th	58.70

4.5.3 Naïve Bayes Classifier

The naïve bays algorithm is one of the data mining classification techniques. This model is a probabilistic model based on the Bayes theorem. With the aforementioned assumption applied to Bayes theorem, the classification is done by calculating the maximum posterior, as mentioned in Chapters Two and Three.

This algorithm has been utilized to classify the dataset by training the model using one of the two chunks that have been generated via the optimization technique (PSO). This is followed by testing the remaining 30% of the dataset separately (testing each 1000 data value). The result of this model in terms of accuracy is 56.78.

Table 4.10: Accuracy results for each 1000 testing dataset using NV classifier

Testing Dataset	**Accuracy results**
1000'1st	54.60
1000'2nd	55.07
1000'3d	55.83
1000'4th	54.34
1000'5th	55.80
1000'6th	56.20
1000'7th	56.07
….	….
Last 1000'th	56.78

Table 4.11: Performance evaluation percent for three classifiers using PSO

Classifier	Accuracy	Precision avg	Recall avg	F1-score avg
Neural network	90.40	0.44	0.46	0.47
SVM	58.70	0.50	0.65	0.56
Naïve Bayes	56.78	0.49	0.50	0.52

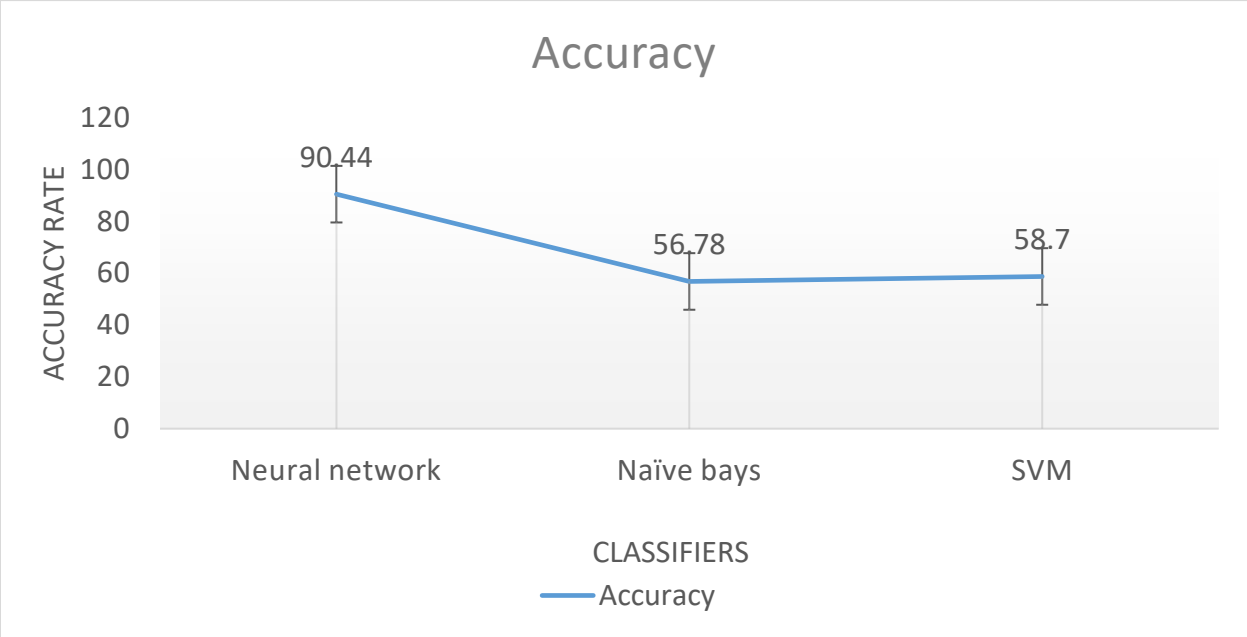

Figure 4.5: Accuracy rate for the three classifiers using PSO algorithm

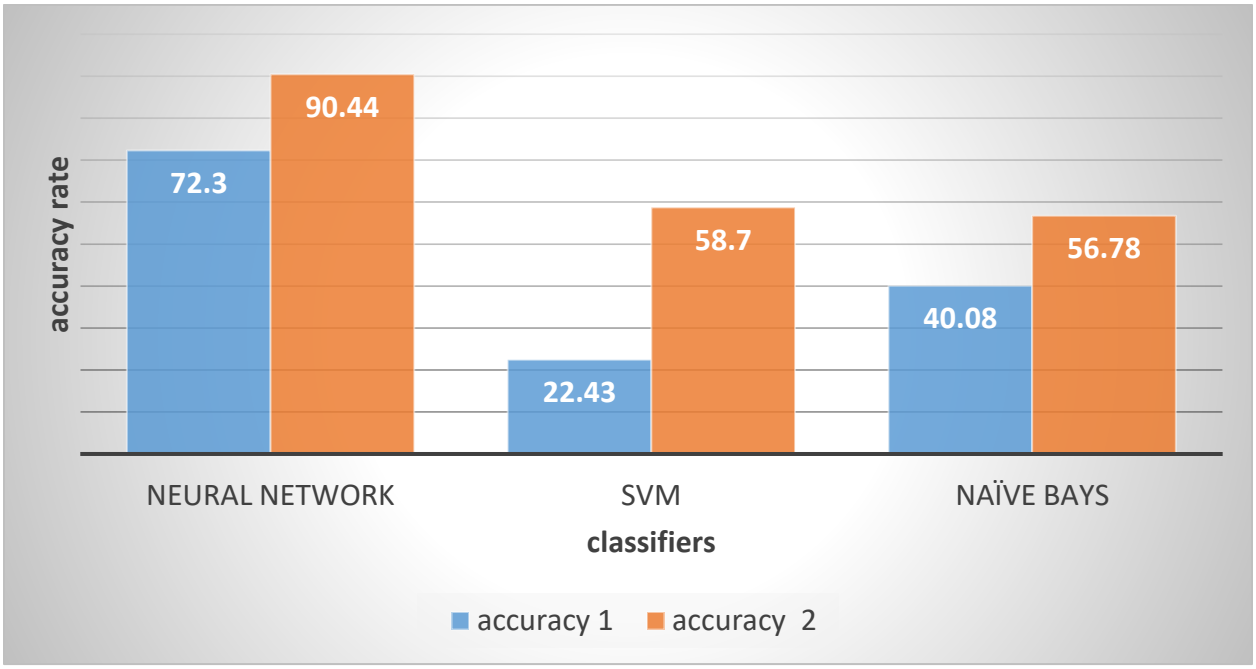

Figure 4.6: Accuracy rate for the three classifiers in the two cases (without PSO, with PSO)

The highest results have been achieved with neural network, as shown in Table (4.11). The other classifiers show a lower accuracy but the overall results reached higher with the use of PSO than without it.

It is clearly shown from the table that the neural network is considered to be a robust classifier with an accuracy rate of 90%. The reason is that the neural networks uses optimum chunks weights generated from PSO. In addition, its hyper parameter tunings such as epoch, training rate, and loss function give best results with larger samples. Despite the small difference between Naïve Bayes and SVM, having rates of about 57% and 59% respectively, the SVM and Naïve Bayes give better accuracy result than without PSO.

Table 4.12: Comparison with the related works

Related works	Tools and techniques	Accuracy results
An Effective Metaheuristic Algorithm for Intrusion Detection System [12]	named search economics with k-means with SVM (SEKS)	80.37%
A meta-heuristic Bayesian network classification for intrusion detection [13]	Bayesian network classification	82.99%
A Feature Selection Model for Network Intrusion Detection System Based on PSO, GWO, FFA and GA Algorithms [18]	PSO, GWO, FFA and GA Algorithms with SVM and J48 ML	79.175%
A Feature Selection Method for Network Intrusion Detection [20]	enhanced krill swarm based on linear nearest neighbor lasso step (LNNLS-KH)	expanded by 10.03%
A Proactive Model for Optimizing Swarm Search Algorithms for Intrusion [9] Detection System [21]	PSO and classification	accuracy has 2% better than PSO and BAT optimization.
The proposed method	**Chunking the dataset using PSO with classification**	**90.44**

4.6 Summary

Overall, it has been noticed from the previous results that the use of PSO has made a significance increase in results as compared to the traditional method whereby no optimization technique is used. In addition, the three models (neural network, SVM, NV) an obtained an increased accuracy, however the first one gained the highest accuracy.

Chapter Five

Conclusions and Future Works

5.1 Conclusion

This thesis provides an effective way to classify the data stream using metaheuristics techniques, namely the PSO algorithm along with data mining techniques. The PSO splits the sensor data into two optimum chunks which are used to train the classifiers based on a mathematical equation. The conclusions drawn on basis of the work presented in this thesis can be summarized through the following points:

1. The test data is evaluated by selecting the best chunks for training these three classifiers and reaching an evaluation.
2. The results show a high accuracy for neural networks.
3. The classifier with the highest accuracy is selected to show the performance of this work, which is the PSO with the neural network, obtaining an accuracy of 90%.
4. The hybrid methods of PSO and neural network enable the use of the mathematical equation to classify such large streaming data from sensors.
5. In addition, it is found to be sufficient to conduct speed while maintaining a high accuracy.
6. The PSO, silhouette metric and mathematical equation contributed to the improvement of the prediction by testing the appropriate chunk.

5.2 Future Work

The future works can be summarized in the following points:

1. The future works include designing a hybrid method using alternative metaheuristic methods such as the Bat algorithm.
2. The model can be developed by changing the datasets to three other ones.
3. The work presented in this thesis could be extended so as to include all classifiers in data mining rather than three classifiers.
4. The model can be enhanced with improving the execution time.

References

[1] V. S. Reddy, T. V Rao, and A. Govardhan, “Data mining techniques for data streams mining,” *A Publ. IIETA*, vol. 4, no. 1, pp. 31–35, 2017, doi: 10.18280/rces.040106.

[2] A. Kumar, “Stream Mining a Review : Tool and Techniques,” pp. 27–32, 2017.

[3] Saxena Saumya, “Basic Concept of Classification (Data Mining),” *GeeksforGeeks*, 2021. https://www.geeksforgeeks.org/basic-concept-classification-data-mining/ (accessed Aug. 28, 2021).

[4] J. Leskovec, A. Rajaraman, and J. D. Ullman, “Mining of Massive Datasets,” *Min. Massive Datasets*, 2020, doi: 10.1017/9781108684163.

[5] B. A. Safae Sossi Alaoui, Yousef Farhaoui, “Classification algorithms in Data Mining,” *Int. J. Tomogr. Simul.*, 2018.

[6] E. Cuevas, E. Barocio Espejo, and A. Conde Enríquez, “Introduction to metaheuristics methods,” *Stud. Comput. Intell.*, vol. 822, no. January, pp. 1–8, 2019, doi: 10.1007/978-3-030-11593-7_1.

[7] S. S. Rao, “Metaheuristic Optimization Methods,” *Eng. Optim. Theory Pract.*, pp. 673–695, 2019, doi: 10.1002/9781119454816.ch14.

[8] X. Li, D. Wu, J. He, M. Bashir, and M. Liping, “An Improved Method of Particle Swarm Optimization for Path Planning of Mobile Robot,” *J. Control Sci. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/3857894.

[9] “Intrusion Detection System (IDS),” *GeeksforGeeks*, 2020. https://www.geeksforgeeks.org/intrusion-detection-system-ids/ (accessed Dec. 12, 2021).

[10] S. Wares, J. Isaacs, and E. Elyan, “Data stream mining: methods and challenges for handling concept drift,” *SN Appl. Sci.*, vol. 1, no. 11, pp. 1–19, 2019, doi: 10.1007/s42452-019-1433-0.

[11] “Iot Device Network Logs ,” *Kaggle*. https://www.kaggle.com/speedwall10/iot-device-network-logs (accessed Dec. 21, 2021).

[12] Z.-H. Chen and C.-W. Tsai, “An Effective Metaheuristic Algorithm for Intrusion Detection System,” *2018 IEEE Int. Conf. Smart Internet Things*, pp. 154–159, 2018, doi: 10.1109/smartiot.2018.00036.

[13] M. K. Prasath and B. Perumal, “A meta-heuristic Bayesian network classification for intrusion detection,” *Int. J. Netw. Manag.*, vol. 29, no. 3, pp. 1–12, 2019, doi: 10.1002/nem.2047.

[14] W. A. H. M. Ghanem and A. Jantan, *A new approach for intrusion detection system based on training multilayer perceptron by using enhanced Bat algorithm*, vol. 32, no. 15. Springer London, 2020.

[15] N. Kunhare, R. Tiwari, and J. Dhar, “Particle swarm optimization and feature selection for intrusion detection system,” *Sadhana - Acad. Proc. Eng. Sci.*, vol. 45, no. 1, 2020, doi: 10.1007/s12046-020-1308-5.
[16] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” *Comput. Networks*, vol. 174, 2020, doi: 10.1016/j.comnet.2020.107247.
[17] H. Mohmmadzadeh, “A Combination Approach of Two Metaheuristic Algorithm for Optimal Feature Selection : Case Study Email Spam Detection,” no. May, pp. 1–25, 2020, doi: 10.20944/preprints202001.0309.v2.
[18] O. Almomani, “A Feature Selection Model for Network Intrusion Detection System Based on PSO, GWO, FFA and GA Algorithms,” *Symmetry (Basel).*, 2020, doi: 10.3390/sym12061046.
[19] W. A. H. M. Ghanem, A. Jantan, S. A. A. Ghaleb, and A. B. Nasser, “An Efficient Intrusion Detection Model Based on Hybridization of Artificial Bee Colony and Dragonfly Algorithms for Training Multilayer Perceptrons,” *IEEE Access*, vol. 8, pp. 130452–130475, 2020, doi: 10.1109/ACCESS.2020.3009533.
[20] X. Li, P. Yi, W. Wei, Y. Jiang, and L. Tian, “LNNLS-KH: A Feature Selection Method for Network Intrusion Detection,” *Secur. Commun. Networks*, vol. 2021, 2021, doi: 10.1155/2021/8830431.
[21] S. S. Alkafagi and R. M. Almuttairi, “A Proactive Model for Optimizing Swarm Search Algorithms for Intrusion Detection System,” *J. Phys. Conf. Ser.*, vol. 1818, no. 1, 2021, doi: 10.1088/1742-6596/1818/1/012053.
[22] L. Rutkowski, M. Jaworski, and P. Duda, “Basic Concepts of Data Stream Mining,” *Stream Data Min. Algorithms Their Probabilistic Prop.*, pp. 13–33, 2020, doi: 10.1007/978-3-030-13962-9_2.
[23] E. Alothali, H. Alashwal, and S. Harous, “Data stream mining techniques: A review,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 2, pp. 728–737, 2019, doi: 10.12928/TELKOMNIKA.v17i2.11752.
[24] U. Kokate, A. Deshpande, P. Mahalle, and P. Patil, “Data stream clustering techniques, applications, and models: Comparative analysis and discussion,” *Big Data Cogn. Comput.*, vol. 2, no. 4, pp. 1–30, 2018, doi: 10.3390/bdcc2040032.
[25] Logsign Team, “Data Stream Clustering Methods Examples ,” *Logsign*, 2020. https://www.logsign.com/blog/data-stream-clustering-methods-examples/ (accessed Aug. 30, 2021).
[26] R. Jadhav and N. Sharma, “Classification methods for Data Streaming Mining,” vol. 3, no. 1, pp. 2456–3293, 2018, [Online]. Available: http://oaijse.com/VolumeArticles/FullTextPDF/203_21._CLASSIFICATION

_METHODS_FOR_DATA_STREAM_MINING.pdf.
[27] E. Garcia-Martin, N. Lavesson, H. Grahn, E. Casalicchio, and V. Boeva, "Hoeffding trees with nmin adaptation," *Proc. - 2018 IEEE 5th Int. Conf. Data Sci. Adv. Anal. DSAA 2018*, pp. 70–79, 2019, doi: 10.1109/DSAA.2018.00017.
[28] F. Stahl, T. Le, A. Badii, and M. M. Gaber, "A frequent pattern conjunction heuristic for rule generation in data streams," *Inf.*, vol. 12, no. 1, pp. 1–26, 2021, doi: 10.3390/info12010024.
[29] Steve Ranger, "What is the IoT?," *ZDNet*, 2020. https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/ (accessed Sep. 15, 2021).
[30] "What Is the Internet of Things (IoT)? ," *SAP Insights*. https://insights.sap.com/what-is-iot-internet-of-things/ (accessed Sep. 25, 2021).
[31] SolarWinds, "What Is an Intrusion Detection System (IDS)? ," *Logical Read*, 2021. https://logicalread.com/intrusion-detection-system/#.YUJAyp0zZPY (accessed Sep. 15, 2021).
[32] "IDS in Security | What is Intrusion Detection System and Functions?," *comodo*. https://www.comodo.com/ids-in-security.php (accessed Sep. 18, 2021).
[33] Pranjal Pandey, "Data Preprocessing: Concepts. ," *Towards Data Science*, 2019. https://towardsdatascience.com/data-preprocessing-concepts-fa946d11c825 (accessed Sep. 11, 2021).
[34] Sadhvi Anunaya, "Data Preprocessing in Data Mining ," *Analytics Vidhya*, 2021. https://www.analyticsvidhya.com/blog/2021/08/data-preprocessing-in-data-mining-a-hands-on-guide/ (accessed Sep. 11, 2021).
[35] Sana Mushtaq, "Data preprocessing in detail ," *IBM Developer*, 2019. https://developer.ibm.com/articles/data-preprocessing-in-detail/ (accessed Sep. 11, 2021).
[36] Jason Brownlee, "A Gentle Introduction to k-fold Cross-Validation," *Machine Learning Mastery*, 2018. https://machinelearningmastery.com/k-fold-cross-validation/ (accessed Jan. 11, 2022).
[37] M. Tennant, F. Stahl, O. Rana, and J. B. Gomes, "Scalable real-time classification of data streams with concept drift," *Futur. Gener. Comput. Syst.*, vol. 75, pp. 187–199, 2017, doi: 10.1016/j.future.2017.03.026.
[38] Victor Roman, "Supervised Learning: Basics of Classification and Main Algorithms | by Victor Roman | Towards Data Science," *Towards Data Science*, 2019. https://towardsdatascience.com/supervised-learning-basics-of-classification-and-main-algorithms-c16b06806cd3 (accessed Sep. 08, 2021).
[39] Mohammad Waseem, "Classification In Machine Learning ," *edureka*, 2021.

https://www.edureka.co/blog/classification-in-machine-learning/ (accessed Sep. 08, 2021).
[40] "Classification Using Neural Networks | by Oliver Knocklein | Towards Data Science." https://towardsdatascience.com/classification-using-neural-networks-b8e98f3a904f (accessed Apr. 26, 2021).
[41] Sidath Asiri, "Machine Learning Classifiers. What is classification?," *Towards Data Science*, 2018. https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623 (accessed Sep. 09, 2021).
[42] Navdeep Singh Gill, "Artificial Neural Networks Applications and Algorithms," *Xenon stack*, 2021. https://www.xenonstack.com/blog/artificial-neural-network-applications (accessed Sep. 09, 2021).
[43] "Support Vector Machine — Introduction to Machine Learning Algorithms | by Rohith Gandhi | Towards Data Science." https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 (accessed May 09, 2021).
[44] R. A. Berk, "Support Vector Machines," vol. 1, pp. 339–359, 2020, doi: 10.1007/978-3-030-40189-4_7.
[45] Rohit Sharma, "Classification in Data Mining Explained," *upGrad* , 2021. https://www.upgrad.com/blog/classification-in-data-mining/ (accessed Sep. 09, 2021).
[46] "Support Vector Machine (SVM) Algorithm," *Javatpoint*. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm (accessed Jun. 30, 2021).
[47] G. Rohith, "Naive Bayes Classifier," *Towards Data Science*. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c (accessed May 09, 2021).
[48] "7 Types of Classification Algorithms ," *Analytics India Magazine*, 2018. https://analyticsindiamag.com/7-types-classification-algorithms/ (accessed Sep. 09, 2021).
[49] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, no. September, 2019, doi: 10.1016/j.cie.2019.106040.
[50] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2191–2233, 2019, doi: 10.1007/s10462-017-9605-z.
[51] N. Gunantara and I. D. N. Nurweda Putra, "The Characteristics of Metaheuristic Method in Selection of Path Pairs on Multicriteria Ad Hoc Networks," *J. Comput. Networks Commun.*, vol. 2019, 2019, doi: 10.1155/2019/7983583.
[52] B. Bruno Seixas Gomes de Almeida (Federal University of Rio de Janeiro,

Rio de Janeiro,), B. Leite, Victor Coppo (Federal University of Rio de Janeiro, Rio de Janeiro, and), “Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems,” in *Swarm Intelligence - Recent Advances, New Perspectives and Applications*, vol. 2, no. 12, J. Del Ser and E. V. and E. Osaba, Eds. 2019, pp. 1–22.

[53] M. Corazza, G. di Tollo, G. Fasano, and R. Pesenti, “A novel hybrid PSO-based metaheuristic for costly portfolio selection problems,” *Ann. Oper. Res.*, vol. 304, no. 1–2, pp. 109–137, 2021, doi: 10.1007/s10479-021-04075-3.

[54] A. S. Ashour and Y. Guo, *Optimization-based neutrosophic set in computer-aided diagnosis*. Elsevier Inc., 2020.

[55] “Silhouette Index – Cluster Validity index ,” *GeeksforGeeks*, 2019. https://www.geeksforgeeks.org/silhouette-index-cluster-validity-index-set-2/ (accessed Sep. 11, 2021).

[56] N. Kaoungku, K. Suksut, R. Chanklan, K. Kerdprasop, and N. Kerdprasop, “The silhouette width criterion for clustering and association mining to select image features,” *Int. J. Mach. Learn. Comput.*, vol. 8, no. 1, pp. 69–73, 2018, doi: 10.18178/ijmlc.2018.8.1.665.

[57] Aayush Bajaj, “Performance Metrics in Machine Learning,” *neptune.ai*, 2021. https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide (accessed Sep. 11, 2021).

[58] Joydwip Mohajon, “Confusion Matrix for Your Multi-Class Machine Learning Model ,” *Towards Data Science*, 2020. https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826 (accessed Sep. 14, 2021).

[59] Aditya Mishra, “Metrics to Evaluate your Machine Learning Algorithm,” *Towards Data Science*, 2018. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234 (accessed Sep. 11, 2021).

جمهورية العراق

وزارة التعليم العالي و البحث العلمي

جامعة بابل

كلية تكنولوجيا المعلومات

كشف البيانات المتدفقة لكشف التسلل بأستخدام ألامثلية و طرق التصنيف

الرسالة

مقدمة الى مجلس كلية تكنولوجيا المعلومات في جامعة بابل جزء من متطلبات نيل درجة الماجستير في تكنولوجيا المعلومات / برامجيات

من قبل

حنين فرحان كاظم سدخان

بأشراف

أ.م.د. مهدي عبادي مانع

٢٠٢١ م ١٤٤٣ هـ

الخلاصة

أن التنقيب عن البيانات المتدفقة تلعب دورًا أساسيًا في تطبيقات الوقت الفعلي. المصادر الرئيسية للبيانات المتدفقة هي أجهزة الاستشعار والوسائط المتعددة والوسائط الاجتماعية. تتسم هذه البيانات بخصائص مميزة تشمل سرعتها العالية وحجمها الضخم للغاية ، بالإضافة إلى قدرتها على التغيير بمرور الوقت. لديها العديد من المشاكل ، أحدها هو مفهوم الانجراف الذي يحدث بسبب الخاصية المستمرة للبيانات المتدفقة. لا تستطيع التقنيات التقليدية للتنقيب عن البيانات التعامل مع هذه البيانات الكبيرة والسريعة أو التنقيب عنها.

الهدف من هذا العمل هو تصنيف بيانات المستشعر بواسطة بناء نظام قوي يمكنه جعل مهمة التصنيف اكثر دقة. يمكن الوصول الى هذا النظام باستخدام الامثلية و طرق التصنيف.

يتم استخدام الامثلية لبناء أجزاء متوازنة من البيانات باستخدام Particle Swarm Optimization (PSO) للحصول على دقة تصنيف أفضل. يُظهر النموذج المشترك الذي تم تكوينه بواسطة الامثلية وطرق التصنيف تحسنًا في أداء الدقة بمعدل مرتفع ، مقارنة بالنماذج التي لا تتضمن الامثلية. تم اختيار عدة مصنفات بناءً على معادلة رياضية لكل جزء لتحديد الأفضل الذي يعطي أفضل النتائج ذات الدقة العالية. أظهرت النتائج التي حصلنا عليها أداء جيد من حيث دقة التصنيف للشبكة العصبية بنسبة 90٪ ومعدل إيجابي منخفض.