JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

Modelling of Hybrid Intrusion Detection System in Internet of Things with Machine Learning Approach

By

Tesfamariam Gashaw

This thesis is submitted to School of Graduate Studies of Jimma University in partial fulfilment of the requirements for the degree of

Master of Science

in

Computer Engineering

August 2020

Jimma, Ethiopia

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

Modelling of Hybrid Intrusion Detection System in Internet of Things with Machine Learning Approach

By

Tesfamariam Gashaw

Advisor: Dr. Getachew Alemu

Co-Advisor: Mr. Fetulhak Abdurrahamn

Submission Date: February, 2020

Declaration

I, the undersigned, declare that this thesis entitled **Modelling of Hybrid Intrusion Detection System in the Internet of Things with Machine Learning Approach** is my original work and has not been presented for a degree in this or any other universities, and all sources of references used for the thesis work have been appropriately acknowledged.

Student Name: Tesfamariam Gashaw **Signature:**_____ **Date:** 20/08/20

Advisor: Dr. Getachew Alemu **Signature:**_____ **Date:** 20/08/2020

Co-Advisor: Mr. Fetulhak Abdurahman **Signature:**_____ **Date:**_____

This Thesis has been Approved by

Board of Examiners:	**Signature:**	**Date:**
External Examiner: Dr. Henock Mulugeta		Aug 17, 2020
Internal Examiner: Mr. Kebebew Ababu		
Chair Person: Mr. Fetulhak Abdurahman		

Acknowledgements

In the beginning, I would like to give all honour and glory to the almighty God for allowing me to do and complete this work. Next, I would like to express my deep and sincere gratitude to my advisers, Dr. Getachew Alemu and Mr. Fetulhak Abdurrahamn for their expert guidance, constructive comments, suggestions, and encouragement without which this thesis work could have not been completed. I am very thankful as well, to my families as they have been a constant source of encouragement throughout the study of this thesis.

Lastly but not least, I would like to thank staff members of the faculty and all peoples stood by my side during the execution period of the thesis work.

Contents

List of Figures

List of Tables

Abbreviations

CNN	**C**able **N**ews **N**etwork
CSV	**C**omma **S**eparted **V**alue
DDos	**D**istrubted **D**enial **O**f **S**ervice
Dos	**D**istrubted **D**enial **O**f **S**ervice
DSL	**D**igital **S**ubscriber **L**ine
HFC	**H**ybrid **F**iber **C**oaxial
IoT	**I**nternet **o**f **T**hings
IDS	**I**ntrustion **D**etection **S**ystem
6LOWPAN	**L**ow **P**ower **P**ersonal **A**rea **N**etwork
ML	**M**achine **L**earning
RPL	**R**outing **P**rotocol for Low Power and **L**ossy Network
SVM	**S**upport **V**ector **M**achine
TCP/IP	**T**ransmission **C**ontrol **P**rotocol **I**nternet **P**rotocol
ROC	**R**eceiver **O**perator **C**haracteristic
Weka	**W**aikato **E**nivromental for **k**nowledge **A**nlysis
WSN	**W**ireless **S**ensor **N**etwork

Abstract

Internet of Things (IoT) devices are interconnected devices that integrate things and the Internet to make human life easy and faster and also Internet of Things devices are interconnected for a longer period without human intervention. This raises to develop security solutions to handle the security issues in the IoT network which is compatible with the services.

In this research, we used raw data to construct the model for the system and after the data is prepared there are different mechanisms that we follow to analyze the data; data pre-processing for removing the irrelevant feature in the data, feature selection for selecting features using random forest algorithm.

We conduct our experiments by selecting four different supervised machine learning for the classification of attacks on the IoT network. From the experimental result cascading two machine learning algorithms (Random forest and Support Vector Machine) performance is better than among other cascading machine learning algorithms.

*Keyword:****IoT,IDS,Random Forest,Support Vector Machine***

Chapter 1

Introduction

1.1 Background Of Study

IoT is a new paradigm that integrates the Internet and physical objects belonging to different domains such as home automation, industrial process, human health, and environmental monitoring. The world is currently witnessing the rapid product launches and high expectations from the emerging IoT technology. It is growing at an accelerating pace connecting billions of devices in our daily life. As per the Gartner event analysis, there will be around 25 billion connected things by the year 2020 [1].

IoT devices are Internet-connected devices in our daily activities, bringing, in addition to many benefits, challenges related to security issues, since the IoT devices have little storage capacity and less processing capacity, and also IoT devices are connected for longer periods without human intervention. This raises a need to develop smart security solutions which are light-weight, distributed and compatible for the services.

For this purpose, the Intrusion Detection System (IDS) is being employed to observe and detect the attacks that may occur on IoT devices networks. Data mining Techniques machine learning, neural networks, collective intelligence, evolutionary

algorithm, and statistical ways are several algorithms that are used for classification, coaching and reviewing detection accuracy with analysis primarily based on the standard datasets in intrusion detection systems.

There are different techniques for intrusion detection mechanisms and some of them describe below:

I. Signature-based approaches.

In signature-based approaches, IDSs detect attacks when a system or network behavior matches an attack signature stored in the IDS internal databases. If any system or network activity matches with stored patterns/signatures, then an alert will be triggered. Signature-based IDSs are accurate and very effective at detecting known threats, and their mechanism is easy to understand. However, this approach is ineffective to detect new attacks and variants of known attacks, because a matching signature for these attacks is still unknown[2].

II. Anomaly-based approaches.

Anomaly-based IDSs compare the activities of a system at an instant against a normal behavior profile and generates the alert whenever a deviation from normal behavior exceeds a threshold. This approach is efficient to detect new attacks, in particular, those attacks related to abuse of resources. However, anything that does not match a normal behavior is considered an intrusion, and learning the entire scope of the normal behavior is not a simple task. Thereby, this method usually has high false-positive rates[3].

III. Specification-based approaches.

The specification is a set of rules and thresholds that define the expected behavior for network components such as nodes, protocols, and routing tables. Specification-based approaches detect intrusions when network behavior deviates from specification definitions. Therefore, specification-based detection has the same purpose of anomaly-based detection: identifying deviations from normal behavior. However, there is one important difference between these methods: in

specification-based approaches, a human expert should manually define the rules of each specification[4].

In this research, the hybrid Intrusion Detection System is proposed by cascading Random Forest and Support Vector Machine learning algorithms to detect the intruders and malicious users effectively and to enhance the accuracy of the system using the proposed systems. Our proposed system has been compared to the existing system and is assumed to allow more accurate detection mechanisms and make a high range of attacks compare to the existing system.

1.2 Statement of the Problem

As the internet users continue to connect more Internet of Things network devices, intruders tend to gain greater surfaces to launch new types of cyber-attacks. This is owed to the fact that the various devices that own constituted the IoT network have little security protection against network-borne threats hence very simple to exploit and intruders can easily exploit poorly protected Internet of Things network devices to cause a varying degree of damages ranging from physical damage, spying, and distributed denial of service.

In October of 2016, the largest DDoS attack was carried out by using IoT botnets. PayPal, The Guardian, Netflix, Reddit, and CNN, particularly, became the target. The botnets were made by a malware called Mirai. This malware exploited the security vulnerability of the Internet of Things device's login information. Exploited devices were directed to the targets. Using default username, password, non-unique passwords and lack of software and firmware updates caused the Mirai attack [5].

Since the Internet of Things network is vulnerable for attackers and different damages are carried out in the IoT network, this is the reason why we decided to work on under this title.

1.3 Objectives of the Research

1.3.1 General Objective

The general objective of this research is to model a hybrid Intrusion Detection System in IoT with machine learning Approach.

1.3.2 Specific Objectives

- To profile some Intrusion Detection techniques in the IoT network then,
- To Specify the requirement of the model
- To develop a model for Intrusion Detection using machine learning algorithm
- To evaluate the performance of the proposed approach using appropriate metrics finally,
- To select best classifier and best method.

1.4 Methodology

To achieve the objective of this thesis first, there is a need for data collection to be capable to construct the model for the system. Collecting data might be costly, therefore, it is possible to use data that is already collected by others from different source to increase the attack detection range to achieve the objective of this research but after the data is collected there is a rigorous data analysis and preparation of the data in the required format before it is used as an input to the model. There are different tools that we used in this research latex for documentation,Anaconda 3.6 for compile the code and Weka 3.8 for train and test the data.

1.5 Scope and Limitation of the Study

This research aims to Model a Hybrid Intrusion Detection System in IoT by combining two machine learning algorithm random forest and Support Vector machine algorithms to achieve of the objective of this thesis and the scope of this research is to cover the proposed detection mechanism for intrusion detection in IoT is designed to be logical control (i.e. software), but not physical control and the capabilities of the model in terms of intrusion detection is limited to detecting anomalies in data at application level not network layer.

1.6 Significance of Thesis

IoT is one of fast growing technology and different devices are interconnected each other to make our life easier,safer and healthier but the intruder get a great surface to attack as the inter connected device network size is large,Therefore,security mechanism is should be implemented to protect the data and privacy from the intruders and this research is significant for a security problem on the IoT by preventing the intruders from accessing the network and to protect the personal privacy information and data from malicious users by monitoring the operation of the networks then alerting the system administrator when it detects a security violation from normal activity.

1.7 Organization of the Study

This thesis report has five chapters. The first chapter is made up of the background of the study, objectives, statement of the problem and methodology of the research. Chapter two presents a review of research works on Intrusion Detection System in IoT by various researchers. Chapter three describes the methodology used, the proposed model to solve the problem proposed in the statement of the problem.Chapter Four describes the result that obtained with different machine learning algorithm and the performance of machine learning algorithm.The final Chapter,Chapter 5 describes the conclusion and the future works.

Chapter 2

Literature Review

This chapter is concerned with the review of the literature. It is divided into two sections.The first section discusses the concept related to the IoT and machine learning.The second section discusses about research works related to Intrusion Detection System in IoT.

2.1 Internet of Things

IoT is one of the fast-growing technologies today.It is technology range from a simple appliance for a smart home to sophisticated equipment for the industrial plant and several types of sensors collect various types of data in the surrounding environment. Generally, IoT has three distinct operation phases: collecting phase, transmission phase, and processing, management and utilization phase[6].

In the Collection phase, the primary objective is to collect data about the physical environment with sensing devices and the devices of the collection phase are usually small and resource-constrained. Communication protocols and technologies for this phase are designed to operate at limited data rates and short distances with constrained memory capacity and low energy consumption.

In the transmission phase, the aim is to transmit the data gathering during the collection phase to applications and users. In this phase, technologies such as Ethernet, WiFi, HFC, and DSL are combined with TCP/IP protocols to build a network that interconnects objects and users across longer distances.

In the processing, management and utilization phase, applications process collect data to obtain useful information about the physical environment. These applications may take decisions based on this information, controlling the physical objects to act on the physical environment. This phase also includes a middleware, which is responsible for facilitating the integration and communication between different physical objects and multi-platform applications [7].
Different alliances, consortiums, special interest groups, and standard development organizations have proposed an overwhelming amount of communication technologies for IoT, which may pose a big challenge for end-to-end security in IoT applications. Most popular technologies for IoT include IEEE 802.15.4, Bluetooth low energy (BLE), WirelessHART, Z-Wave, LoRaWAN, 6LoWPAN, RPL, CoAP, and MQTT (Message Queue Telemetry Transport). IEEE 802.15.4 is a standard proposed by the Institute of Electrical and Electronics Engineers (IEEE) for physical and medium access control layers of low-rate wireless personal area networks. With the IEEE 802.15.4, devices can operate with data rates from 20 kbps to 250 kbps and transmission ranges from 10 m to 100 m. Medium access control uses the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) technique[8].

2.1.1 IoT Architecture

In an IoT architecture, each layer is defined by its functions and the devices that are used in the layer. There are different opinions regarding the number of layers in IoT. However, according to many researchers [9,12], the IoT primarily operates on three layers which are the Perception, Network, and Application layer. Each layer of IoT has inherent security issues associated with it. Fig.2.1 shows the basic

three-layer architectural framework of IoT concerning the devices and technologies that encompass each layer.

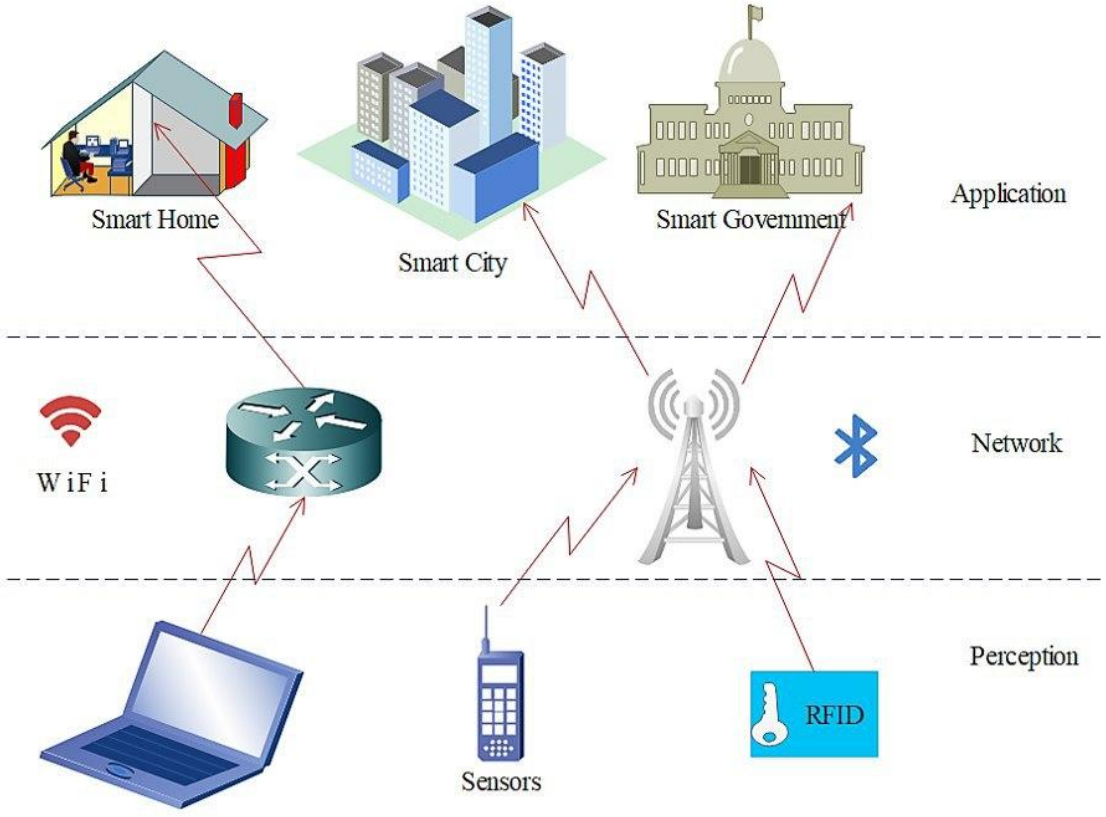

FIGURE 2.1: IoT Architecture

1. Perception Layer

The perception layer is also known as the "Sensors" layer in IoT. The purpose of this layer is to acquire data from the environment with the help of sensors. This layer detects, collects, and processes information from sensors and then transmits it to the network layer. Also, this layer may perform IoT node collaboration in local and short-range networks [11].

2. Network Layer

The network layer of IoT serves the function of data routing and transmission to different IoT hubs and devices over the Internet. At this layer, Internet gateways, switching, and routing devices, etc. operate by using some of the very recent technologies such as WiFi, LTE, Bluetooth, 3G, Zigbee, etc. to provide heterogeneous

network services. The network gateways serve as the mediator between different IoT nodes by aggregating, filtering, and transmitting data to and from different sensors [12].

3. Application Layer

The application layer guarantees the authenticity, integrity, and confidentiality of the data. At this layer, the purpose of IoT which is the creation of smart environments is achieved.

2.1.2 Security Challenges in IoT

Along with the rapid growth of technology in computer networks such as IoT, security has become a critical challenge. The main security requirements for the IoT are as follows [10]: a) data confidentiality and authentication and b) privacy and trust among users and things. The communication in the IoT can be secured by using standard mechanisms such as cryptography and authentication techniques; however, these preventive mechanisms cannot detect all possible attacks, because of the nature of wireless communication. On the other hand, the resource-constrained devices are directly connected to unreliable Internet via IPv6 and 6LoWPAN networks in the IoT; so, they are vulnerable to intrusions (both from the Internet and WSNs) [13].

Each IoT layer is susceptible to security threats and attacks. These can be active, or passive, and can originate from external sources or internal networks owing to an attack by the Insider [14]. The active attack directly stops the service while the passive kind monitors IoT network information without hindering its service. At each layer, IoT devices and services are susceptible to Denial of Service attacks(DoS), which make the device, resource, or network unavailable to authorized users. The security issues at each layer explain below:

1. Perception Layer

There are three security issues in the IoT perception layer. First is the strength of wireless signals. Mostly the signals are transmitted between sensor nodes of IoT

using wireless technologies whose efficiency can be compromised by disturbing waves. Secondly, the sensor node in IoT devices can be intercepted not only by the owner but also by the attackers because the IoT nodes usually operate in external and outdoor environments, leading to physical attacks on IoT sensors and devices in which an attacker can tamper the hardware components of the device. The third is the inherent nature of network topology which is dynamic as the IoT nodes are often moved around different places. The IoT perception layer mostly consists of sensors and RFIDs, due to which their storage capacity, power consumption, and computation capability are very limited making them susceptible to many kinds of threats and attacks [14,15].

The confidentiality of this layer can easily be exploited by Replay Attack which can be made by spoofing, altering, or replaying the identity information of one of the devices in IoT. Or the attacker might gain the encryption key by analyzing the required time to perform the encryption that is known as Timing Attack. Another confidentiality threatening attack is when the attacker takes over the node and captures all information and data which is the Node Capture attack. The attacker can add another node to the network that threatens the integrity of the data in this layer by sending Malicious Data. This can also lead to a DoS attack, by consuming the energy of the nodes in the system and depriving it of the sleep mode that the nodes use to save the energy [16].

2. Network Layer

As mentioned before, the network layer of IoT is also susceptible to DoS attacks. Apart from the DoS attacks, the adversary can also attack the confidentiality and privacy at the network layer by traffic analysis, eavesdropping, and passive monitoring [14]. These attacks have a high likelihood of occurrence because of the remote access mechanisms and data exchange of devices. The network layer is highly susceptible to the Man-in-the-Middle attack, which can be followed by eavesdropping. If the keying material of the devices eavesdrops, the secure communication channel will be completely compromised. The key exchange mechanism in IoT must be secure enough to prevent an intruder from eavesdropping and then committing identity theft. The communication in the IoT is different than that of

the Internet because it is not restricted to machines to humans. However, the feature of machine-to-machine communication that the IoT introduces has a security issue of Compatibility. The heterogeneity of the network components makes it difficult to use the current network protocols as is, and still produce efficient protection mechanisms. Attackers can also take advantage of the fact that everything is connected to gain more information about the users and use this information for future criminal activities [9].

Protecting the network is important in the IoT, but also protecting the objects in the network is equally important. Objects must have the ability to know the state of the network and the ability to protect themselves from any attacks against the network. This can be achieved by developing protocols as well as software that enables objects to respond to any situations and behaviors that can be considered abnormal or may affect their security [17].

3. Application Layer

Since the IoT still does not have global policies and standards that govern the interaction and the development of applications, there are many issues related to security. Different software and applications have different authentication mechanisms, which makes integration of all of them very difficult to ensure data privacy and identity authentication. The large amounts of connected devices that share data will cause large overhead on applications that analyze the data, which can have a big impact on the availability of the services. One more issue that must be considered when designing the applications in IoT is how different users will interact with them, the amount of data that will be revealed, and who will be responsible for managing these applications. The users must have some tools to control what data they want to disclose and they must be aware of how the data will be used, by whom, and when.

2.2 Machine Learning

Machine Learning is a field of computer science that is concerned with making a computing machine capable of learning without direct programming or intervention from humans. In 1997, Tom Mitchell – the Chair of ML at Carnegie Mellon University, provides a definition of ML which is widely used nowadays. The definition is: "A computer program is said to learn from experience E for some class of tasks T and performance P, if its performance at tasks in T, as measured by P, improves with the experience E".

In Mitchell definition of ML, three objects can be observed;

1. Task (T) – can be one or more
2. Performance (P)
3. Experience (E)

It can be said that a computer program that runs a set of tasks is learning if the performance of the tasks being conducted by this program improves based on the experience. Therefore, the relation between experience and performance can be described as a linear relation. Therefore, more experience means improved performance. Therefore, a system that can accept previous data as an input, harness and analyze this data to systematically to find patterns, and then undertake predictions is a system that is capable to learn with continuous learning and tuning, the predictions that the system can produce become more fine and accurate [18].

The Emphasis of machine Learning is on automatic methods, and the goal is to build learning algorithms that do the learning automatically without human interference or help. It uses computational methods to learn information directly from data without depending on a predetermined equation as a model. The algorithms find natural patterns in data that generate insight and help to make better decision and prediction [19].

Learning styles in Machine Learning Although "learning" is very important in ML, it is not the goal. According to Shalev-Schwartz and Ben-David (2014), the

main goal of machine learning is to produce a system that can automatically and correctly detect meaningful patterns in data.

2.2.1 Criteria for Selection of Machine Learning Algorithm

The choice of which specific learning algorithm we should use is a critical step. The classifier's evaluation is most often based on prediction accuracy (the percentage of correct prediction divided by the total number of predictions). There are at least three techniques that are used to calculate a classifier's accuracy. One technique is to split the training set by using two-thirds for training and the other third for estimating performance. In another technique, known as cross-validation, the training set is divided into mutually exclusive and equal-sized subsets and for each subset, the classifier is trained on the union of all the other subsets. The average error rate of each subset is therefore an estimate of the error rate of the classifier. Leave-one-out validation is a special case of cross-validation [20].

All test subsets consist of a single instance. This type of validation is, of course, more expensive computationally, but useful when the most accurate estimate of a classifier's error rate is required. If the error rate evaluation is unsatisfactory, a variety of factors must be examined: perhaps relevant features for the problem are not being used, a larger training set is needed, the dimensionality of the problem is too high, the selected algorithm is inappropriate or parameter tuning is needed. A common method for comparing supervised ML algorithms is to perform statistical comparisons of the accuracies of trained classifiers on specific datasets (Bouckaert 2003).

For this research, we select four supervised machine learning algorithms for training and test the data. The selection of these four supervised machine learning algorithms based on the data set that are prepared to model the system and, in a supervised machine learning algorithm those four-machine learning algorithms are mostly used for classification tasks and it has a good accuracy result.

2.3 Supervised Learning

The objective of supervised machine learning is to construct a model that predicts based on Facts when there is uncertainty. A supervised learning algorithm uses a known set of input data and known output or response to the input data then it trains a model to generate reasonable predictions as a response to new data. Supervised learning uses classification and regression techniques. We select four supervised machine learning algorithm for this :

2.3.1 Naive Bayes Algorithm

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given sample belongs to a particular class. The Bayesian classifier is based on Bayes' theorem. Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computation involved and, in this sense, is considered "Naive".

In simple terms, a Naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for Naive Bayes models uses the method of maximum likelihood; in other words, one can work with the Naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

Despite their Naive design and over-simplified assumptions, Naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, analysis of the Bayesian classification problem has shown that there are some theoretical reasons for the unreasonable efficacy of Naive Bayes classifiers.[21] Still, a comprehensive comparison with other classification methods in 2006 showed that Bayes

classification is outperformed by more current approaches, such as boosted trees or random forests.[18] An advantage of the Naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

Bayes Theorem:

Bayes theorem is stated as the probability of the event **B** given **A** is equal to the probability of the event**A** given **B** multiplied by the probability of **A** upon the probability of **B**.

$$P(A/B) = \frac{P(B/A) * P(B)}{P(A)} \tag{2.1}$$

Where as, *P(A/B)*: probability (conditional probability) of occurrence of event A given the event B is true.

P(A): the probability of the occurrence of event A.

P(B): the probability of the occurrence of event B.

P(B/A): the probability of the occurrence of event B given the event A is true.

2.3.2 Decision Tree Algorithm

A decision tree is a classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called ”root” that has no incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is called an internal or test node. All other nodes are called leaves (also known as terminal or decision nodes). In a decision tree, each internal node splits the instance space into two or more sub-spaces according to a certain discrete function of the input attributes values. In the simplest and most frequent case, each test considers a single attribute, such that the instance space is partitioned according to the value of the attribute. In the case of numeric attributes, the condition refers to a range.

Each leaf is assigned to one class representing the most appropriate target value. Alternatively, the leaf may hold a probability vector indicating the probability of the target attribute has a certain value. Instances are classified by navigating them from the root of the tree down to a leaf, according to the outcome of the tests along the path. Figure 2.2 describes a decision tree that reasons whether or not a potential customer will respond to a direct mailing. Internal nodes are represented as circles, whereas leaves are denoted as triangles. Note that this decision tree incorporates both nominal and numeric attributes. Given this classifier, the analyst can predict the response of a potential customer (by sorting it down the tree) and understand the behavioral characteristics of the entire potential customer population regarding direct mailing. Each node is labeled with the attribute it tests, and its branches are labeled with its corresponding values.

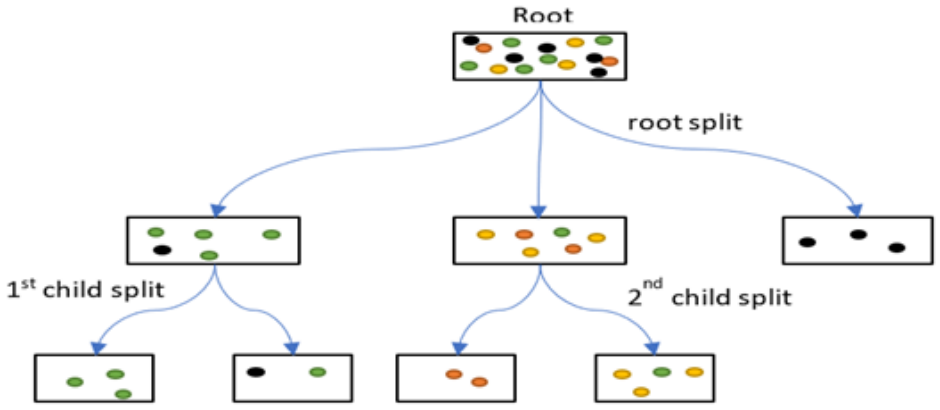

FIGURE 2.2: Figure For Decision Tree

2.3.3 Random Forest Algorithm

Random Forest is one of the supervised machine learning algorithms that work through a bagging approach to create a bunch of decision trees with a random subset of the data. It is considered to be one of the most effective algorithms to solve almost any prediction task. It can be used both for classification and the regression kind of problems. It is a combination of tree predictors where each tree

depends on the values of a random vector sampled independently with the same distribution for all trees in the forest.

Random Forest Machine Learning Algorithm maintains accuracy even when there is inconsistent data and is simple to use. It also gives estimates of what variables are important for the classification. It runs efficiently on large databases while generating an internal unbiased estimate of the generalization error. It also provides methods for balancing error in class population unbalanced data sets but analyzing them theoretically is difficult and the formation of a large number of trees can also slow down prediction while handling real-time systems. There is also another drawback that is, it does not predict beyond the range of the response values in the training data.[22]

2.3.4 Support Vector Machine Algorithm

In today's usage of machine learning, Support Vector Machine is thought in a concert of the strongest and most correct ways within the machine learning algorithms. SVM is among the supervised learning ways that are used for classification, prediction, and regression. This technique is a comparatively new approach that in recent years has shown sensible performance for classification compared to older ways like perception neural networks and it is easy as well. SVM because of its sensible ability to generalize and being superior to different algorithms in classification and regression is extremely widespread. SVM, in theory, is intended for binary classification; therefore, its method forward to solve the existing classification issues between traditional and abnormal or suspicious behavior is appropriate in follow pattern audit [22]. SVM base on decision planes concept to define the boundaries. Hyperplanes are constructed in a high dimensional space with an SVM, separating all the data points of one category from the other. The hyperplane with the largest margin between the two classes for an SVM is the best. It mostly happens that classes cannot be separated linearly. Due to this, the original finite-dimensional space is mapped into a much higher-dimensional space, which makes separation easier by using what is referred to as "kernel trick".

Though we can apply SVM to a variety of optimization problems like regression, the most challenge is that of classification of data. SVM identifies the data points as being negative or positive, and the problem is to find an optimal hyper-plane separating the data points with a higher margin.

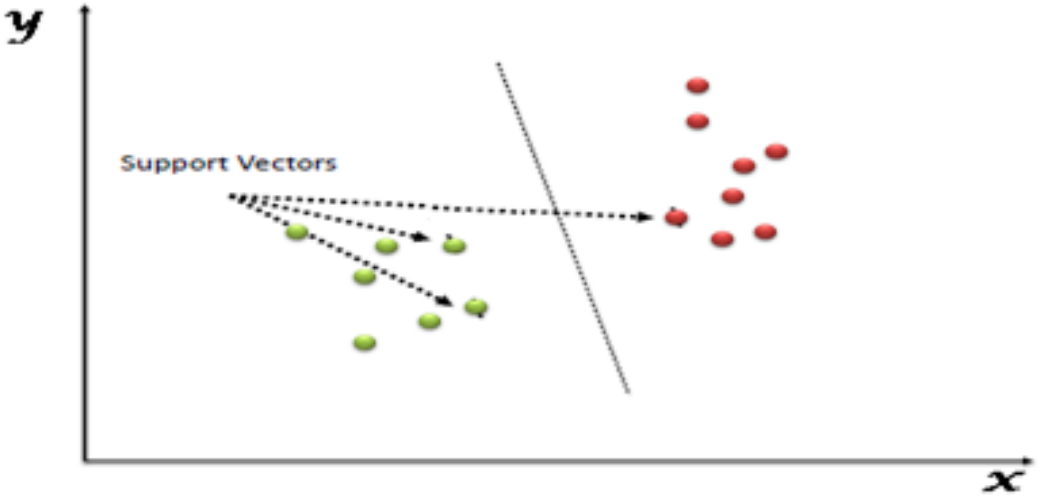

FIGURE 2.3: Figure For Support Vector Machine

Support Vectors are simply the coordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

2.4 Unsupervised Learning

Unlabeled data is used in unsupervised learning and finds hidden patterns or vital structures in data. It is used to draw inferences from datasets that consist of input data without labeled responses. The objective of unsupervised learning is to model the underlying structure or distribution in the data to learn more about the data.

Clustering is a widely used unsupervised learning technique. It is used to find hidden patterns or for exploratory data analysis to group data. Gene sequence analysis, market research, and object recognition can be mentioned in the application areas of clustering.

2.5 Reinforcement Learning

labeled and unlabeled data can be used in reinforcement learning to form basic knowledge. Reinforcement learning works by having the teacher rewarding the system for each correct or incorrect prediction. The reward acts as feedback that the system can rely on while making the next predictions. In time, the system will be able to have access to a knowledge base full of execution paths based on the input used to be given to the system. When a new input is given to the system, the system will try to find the best execution path or combine more than one execution path to produce predictions and wait for the reward. If the reward happens to be better than the previous rewards regarding the same input, then this path becomes favorable. Reinforcement learning is used in online games such as chess. When the machine plays chess.

2.6 Related Works

Over recent years, several works have been done on Intrusion Detection Systems for technologies related to the Internet of Things. In this section, we conduct a literature review related to the Intrusion Detection System in the Internet of Things.

In 2015, Cervantes et al. [23] proposed an Intrusion Detection System for the Internet of Things to detect Sinkhole attack in 6LoWPAN in the Internet of Things. The placement strategy followed was a distributed system since it used a hierarchical structure of nodes. Each node as a role in the system and the main task is to monitor a superior node estimating its traffic patterns. The approach combines concepts of trust and reputation in a specification-based method with an anomaly-based method to monitor the exchange of packets between nodes. When a node detects a sinkhole attack, it broadcasts a message to alert the other nodes but with their proposed approach and the result shows that they effectively detect Sinkhole attack in 6LoWPAN in the Internet of Things but in this system, it can only detect the sinkhole attacks in the Internet of Things.

In 2016, Thanigaivelan et al. [24] present a hybrid Intrusion Detection System for the Internet of Thing. Their approach assigns different tasks to the network nodes and the border router, forcing them to work cooperatively. Each node as an Intrusion Detection System module to monitor their neighborhoods and to send notifications of possible attacks to the Intrusion Detection System module on the border router. The Intrusion Detection System module in the border router receives the notifications from the nodes and decide if there were an intrusion or not. The anomaly-based method consists of looking for deviations of normal behavior learned from the monitoring information, but the authors did not provide many details about the method of determining the normal behavior .

In 2017, S. Prachi [25] presents a novel lightweight Intrusion Detection System for RPL based 6LoWPAN networks to detect the wormhole attacks on the Internet of Thing. In this paper, they used three machine learning algorithms to detect

the wormhole attack on the Internet of Thing and this is the first time machine learning has been used to develop the Intrusion Detection System on the IoT. Finally, the result shows that they effectively detect the wormhole attack using the proposed machine learning algorithms and the performance for K-means approach achieves 70-93% detection rate for varying sizes of random IoT networks, Decision tree-based IDS achieves 71-80% detection rate and the hybrid approach attains 71-75% detection rate for the same network sizes but this system can only detect the wormhole attack in the IoT.

In 2018, K. John and B. James [26] present hybrid Intrusion Detection System in the Internet of Things using Support Vector Machine and decision tree, in this paper new algorithm is proposed by cascading decision tree and Support Vector Machine algorithm to improve classification of attacks and consequently the security systems. The result shows that they can effectively detect different attacks and also, they improve the prediction speed and training time. The proposed algorithm still has some weaknesses such as the classification accuracy and the accuracy of the proposed algorithm, even though decreased in a small percentage, further studies can be carried out to improve the proposed algorithm in terms of accuracy.

In 2018 S. Chawla and G. Thamilarasu [27] presents Real-time Intrusion Detection in the Internet of Things and this paper, proposes a novel intrusion detection system that uses deep learning algorithms to detect security anomalies in the Internet of Thing networks. This detection platform provides security as a service and facilitates interoperability between various network communication protocols used in Internet of Thing and the result show that they effectively detect some attack in the Internet of Thing network but this system has weakness such as further strengthening the machine-learning detection module to enable more accurate identification of attacks in the Internet of Things network.

In 2018 T. A. Mohamed et al. [28] presents machine learning-based Intrusion Detection in Internet of Thing and this paper shows that a proposed Intrusion Detection System based on machine learning techniques to be implemented into

the Internet of Things platforms as a service and they used Random forest algorithm as a classifier to detect intrusions, then applied neural network classifier to detect the categorization of the detected intrusion. The experimental results showed that the proposed model can effectively detect intrusions in the Internet of Thing but the proposed system has low detection accuracy and the techniques include further strengthening the machine-learning detection module to enable more accurate identification of anomalies in an Internet of Things network and should be improved in the future.

We already discuss different related work to this research but still different types of attacks carried out in the IoT because a large number of IoT devices are connected and the size of the network also large. Currently, the security mechanisms in the IoT target only a specific number of attacks, week detection method, and the attack detection range is small. To fill the gap and limitation of currently available intrusion detection in the IoT, we developed a hybrid intrusion detection in IoT to detect a different number of attacks in the IoT and to increase the attack detection range.

Chapter 3

System Designing and Modelling

3.1 Overview

The main objective of this chapter is to design the general architecture of the study to help for proper result implementation and interpretation. To do so; this study will go through multiple steps and we will discuss each step in detail and draw diagrams accordingly under this chapter. This section briefly describes the methodology used to achieve the objectives of the research. It gives a detail explanation of the raw data used in this research and how-to analysis, preprocesses, and feature selection mechanisms before feeding the data to the machine learning algorithms.

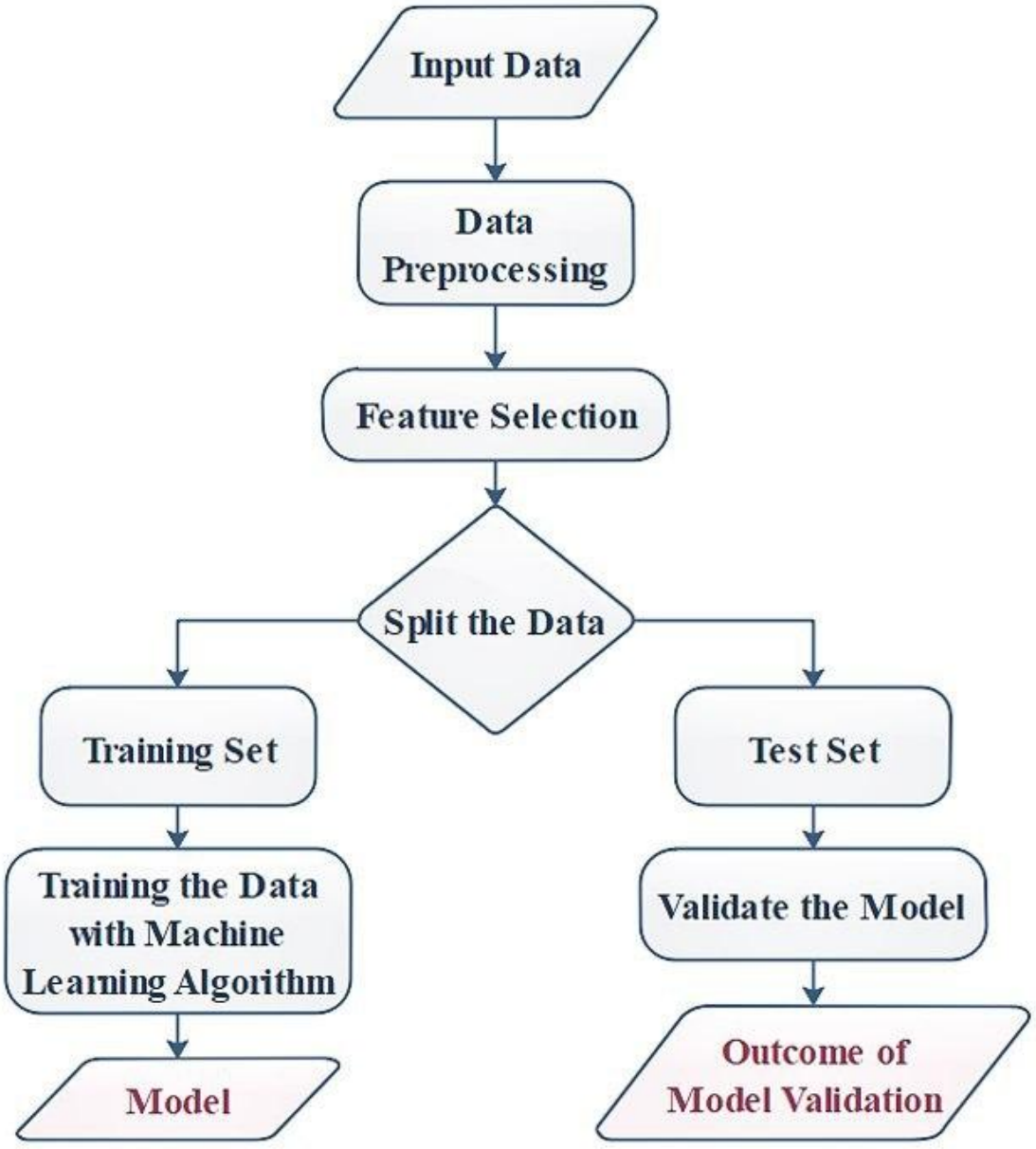

FIGURE 3.1: Overall System Design

The above figure describes the overall architecture of the study and we will discuss in detail the overall design of the system.

3.2 Data Preparation

One of the major research challenges in network intrusion detection systems is the unavailability of a compressive data set which can reflect modem network scenarios. There are different data set for evaluating the Network Intrusion detection system those are KDD98, KDDCUP99, and NSL-KDD but this data set is generated a decade ago and it doesn't represent modern normal and attacks activities. The paper that we stated in the literature review" Modelling hybrid Intrusion Detection System in the Internet of Things using Support Vector Machine and decision tree "used NSL-KDD data set for evaluating the network but the data set is outdated and the attack that incorporates in this dataset is small.

The data set that we used in this research contains real modern attack and normal activities of the network traffic and the data set is contains millions of records for network traffic.The first step in preparing the data set is collecting the data that contains modern normal and different attack types on the network and the data contains different features that are different types: Integer, Float, Binary, Nominal(textual data), and Timestamp values. The address for the raw data to construct the model for the system [31][32].

TABLE 3.1: Types of attacks in IoT

Type of attack	Number of records	Description
Normal	2,218,716	Normal transaction data.
Fuzzers	24,246	An attack in which the attackers attempt to discover security loopholes in application,operating system and it makes random data crash.
Analysis	2,677	It contains different attacks of the port scan,spam and HTML files penetrations.
Backdoors	2,329	It is a technique of bypassing a stealthy normal authentication, securing unauthorized remote access to a device.
Dos	16,353	A malicious attempt to make a server or a network resource unavailable to users.
Exploits	44,525	The attacker knows of a security problem within an operating system or a piece of software and leverages that knowledge by exploiting the vulnerability.
Generic	215,481	Is a technique that establishes against every block-cipher using a hash function to cause a collision without respect to the configuration of the block cipher.
Reconnaissance	13,987	Also can be defined as a probe, and it is an attack which gathers information about a computer network to evade its security controls.
Shellcode	1,511	Is malware in which the attacker penetrates a slight piece of code starting from a shell to control the compromised machine.
Worms	174	Is an attack in which the attacker replicates itself to spread on other computers.Often, it utilizes a computer network to spread itself, depending on the security failures of the target computer used to access it.

As shown above, the data set contains different modern attack in the Internet of Things network and the normal flow of network and the first task is accomplished by preparing the data set for the Internet of Thing network then the data set is ready for the next phase of process.

3.3 Data Preprocessing

Data pre preprocessing is a data mining technique that deals with transforming the raw data to the required format to handle the raw data and after preparing the dataset, the data is read from CSV(Comma Separated Value) files with the panda's module. The data prepared to make it suitable for machine learning, to cleaning the data to remove incomplete variables and to sampling the data further to reduce running times for algorithms and memory requirements.

The raw data contains numerical and nominal(textual data). The first phase that we make in the preprocessing is concatenating CSV file by analyzing the raw data and split the raw data that are concatenating CSV file into different feature types. The feature in the raw data contains different numerical and nominal data then, we convert the features to their corresponding values and remove the irrelevant features in the raw data by selecting a subset of significant features that fully represent the problem.

After converting the features to their corresponding value the next phase is to replace "NAN"(Not A Number) by analyzing the raw data and vectorizing nominal data to binary vectors then, we remove' NAN if any' in the raw data and this value has no impact on the final model and it makes the model complex. Finally, we normalizing the data by changing the value to"0" and "1" for the next phase of the process.

The preprocessing phase is accomplished then the data is saved in HDF5 format to store large amount data and the data is prepared to the next phase of process.

3.4 Feature Selection

It is the process of automatic selection of attributes in the training data that are most relevant to the predictive modeling problem. Feature selection is different from dimensionality reduction. Both methods seek to reduce the number of attributes in the dataset, but a dimensionality reduction method does so by creating new combinations of attributes, whereas feature selection methods include

and exclude attributes present in the data without changing them. It can be used to identify and remove irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model or may decrease the accuracy of the model. The selection criteria are depending on the probabilities of the respective attribute in the class label values.

In feature selection, the first phase we make is taking a sample for training and the maximum number of record for a training sample is 5000 then we remove the irrelevant features and those features are not any influence on the output and having irrelevant features in the data can decrease the accuracy of the models and it makes the model complicated [33].After removing the irrelevant features on the data the next phase is removing redundant features by analyzing the data and these data are an impact on the accuracy of the final model.

Feature importance is the technique used to select features on the data and for selecting features we used a random forest algorithm due to it has good accuracy, robust, and easy of use. The criteria for selecting features on the data are based on top ranking features and we selected 20 features on the data.

Feature selection phase is accomplished by removing the irrelevant feature from the data and the data is prepared for the next phase of the process.

3.5 Splitting the Data

The Feature selection phase is accomplished then, the data is split into a training set and test set. The training set contains a known output and the model learns on this data to be generalized to other data later on and the test set is preparing to test our model's prediction on this test data set. There is a different way of split the data set into training and test set from different techniques I select 70/30 way of splitting the data set into a training set and test set by 70% is prepare for the training set and 30% is prepared for the test set.

A new algorithm is proposed by cascading random forest and Support Vector Machine algorithms to improve the classification of attacks. The proposed algorithm used the random forest for selecting features in the data and Support Vector Machine for classification of attacks due to its good performance when it compares to other machine learning algorithm.

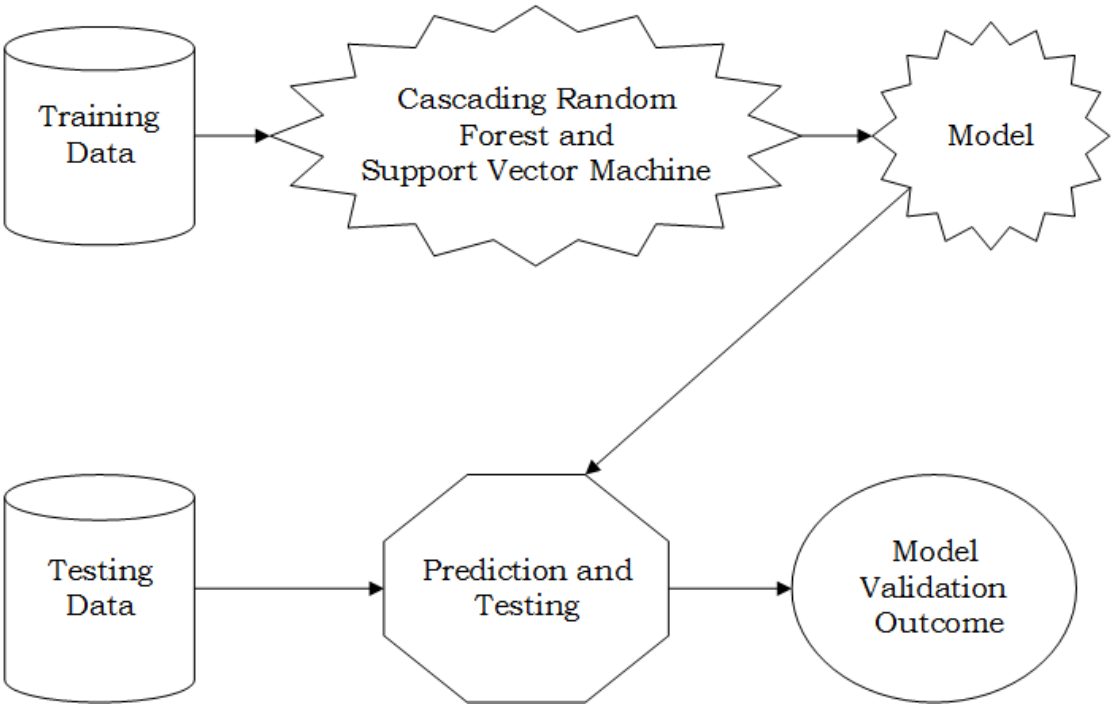

FIGURE 3.2: Model Design

Splitting the data set is accomplished by preparing the data set into training data and test data as shown above in the figure splitting the data set is prepared to the next phase of the process.

The above figure shows the model for the system after train the data with different machine learning algorithms (Naive Bayes, decision tree, random forest, and Support Vector Machine)to get the model for the system. After getting the model for the system the next phase of the process is to validate the model with fresh data(testing data) with different performance metrics.

3.6 Tool for Train and Test Data

WEKA is selected for train and test data. It is a data mining system developed by the University of Waikato in New Zealand that implements data mining algorithms. WEKA is a state-of-the-art facility for developing machine learning (ML) techniques and their application to real-world data mining problems. It is a collection of machine learning algorithms for data mining tasks. The algorithms are applied directly to a dataset. WEKA implements algorithms for data preprocessing, classification, regression, clustering, association rules; it also includes visualization tools. The new machine learning schemes can also be developed with this package. WEKA is open source software issued under the GNU General Public License [34].

Chapter 4

Result and Discussion

4.1 Overview

In this chapter, we will discuss in detail how to train the data that are prepared for the machine learning algorithm and test the data by different machine learning algorithms to select the best performance machine-learning algorithm to achieve best performed. In this research, we select four supervised machine learning algorithms Naive Bayes algorithm, Decision tree algorithm, Support Vector Machine algorithm, and Random forest algorithm to train the data with different machine learning algorithms,and finally, we select best-performed algorithm among them.

4.2 Training the Data

The model builder processes start from reading the arff file format files from the directory. The WEKA classifier splits the instances using cross-validation split as a training and testing set. The cross-validation split gets the input of instances and the number of folds that are going to be trained and tested. Then we have a two-dimensional array of instances type for n number of folds iterate over them, split the instances for training and testing set using the trainCV and testCv methods of WEKA instance class methods. We have an array of models we call them

the WEKA classifiers. Now, the classifier accepts an input of selected algorithms, training set splits and testing set splits.

The data is loaded into WEKA, the next phase is a select learning algorithm. Classifiers in WEKA are the models for predicting nominal or numeric quantities.The learning schemes available in WEKA include decision trees and lists, instance-based classifiers, Support Vector Machines, random forest, multi-layer perceptrons, logistic regression, and Bayes' nets."Meta"-classifiers include bagging, boosting, stacking, error-correcting output codes, and locally weighted learning [35].

From the available learning algorithm in WEKA, we select four(4)supervised machine learning algorithms (Naive Bayes, Decision Trees, Random Forest, and Support Vector Machine) for training and testing the data in WEKA.

4.3 Naive Bayes Algorithm

4.3.1 Performance Evaluation

The efficiency of any machine learning is determined using measures such as true positive rate, false-positive rate, confusion matrix, ROC curve true negative rate, and false-negative rate. Training and evaluating statistical performance on the same data yields over-optimistic results. Cross-validation improves performance accuracy. The train set is used to train the data and test set is used to validate the model.

After selecting the learning algorithm in WEKA it needs set test options for the specified learning algorithm. The test options under WEKA are the following:

1. Use the training set. Evaluates the classifier on how well it predicts the class of the instances it was trained on.

2. Supplied test set. Evaluates the classifier on how well it predicts the class of a set of instances loaded from a file. Clicking on the 'Set. . . ' button brings up a dialog allowing you to choose the file to test on.

3. Cross-validation. Evaluates the classifier by cross-validation, using the number of folds that are entered in the 'Folds' text field. In k-fold cross-validation, data is partitioned into k equal size folds. The k iterations are trained and validated such that within each iteration, the different fold of the data is held out for validation and the remaining k-1 fold is used for learning.

4. Percentage split. Evaluates the classifier on how well it predicts a certain percentage of the data, which is held out for testing. The amount of data held out depends on the value entered in the '%' field.

To calculate the accuracy the Evaluation classes of WEKA is used in which we collect every group of predictions for current model in a Fast Vector. Finally, the model builder builds a model and returns summary of training testing pair using the Evaluation object of summary string method, class detail string and matrix string methods.

Time is taken to build the model: 1.22 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 14.65 seconds

=== Summary ===

Correctly Classified Instances: 96150 54.836 %

Incorrectly Classified Instances: 79191 45.164 %

Kappa statistic: 0.4697

Mean absolute error: 0.0904

Root mean squared error: 0.295

Relative absolute error: 57.1768 %

Total Number of Instances: 175341

=== Detailed Accuracy by Class ===

True Positive Rate (Sensitivity): True Positive Rate is defined as TP/ (FN+TP). True Positive Rate corresponds to the proportion of positive data points that are

correctly considered as positive, with respect to all positive data points.

$$TruePositive\,rate = \frac{TruePositive}{FalseNegative + TruePositive} \tag{4.1}$$

False Positive Rate (Specificity): False Positive Rate is defined as FP / (FP+TN). False Positive Rate corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points.

$$FalsePositive\,rate = \frac{FalsePositive}{FalsePositive + TrueNegative} \tag{4.2}$$

Precision: It is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$Precision = \frac{TruePositives}{TruePositives + FalsePostitives} \tag{4.3}$$

Recall: It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{4.4}$$

F Score is used to measure a test's accuracy. F Score is the Harmonic Mean between precision and recall. The range for F Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances). High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as:

$$FScore = 2 * + \frac{1}{1/Precision + 1/Recall} \tag{4.5}$$

ROC is the area under the curve of plot False Positive Rate Vs True Positive Rate at different points in [0, 1].

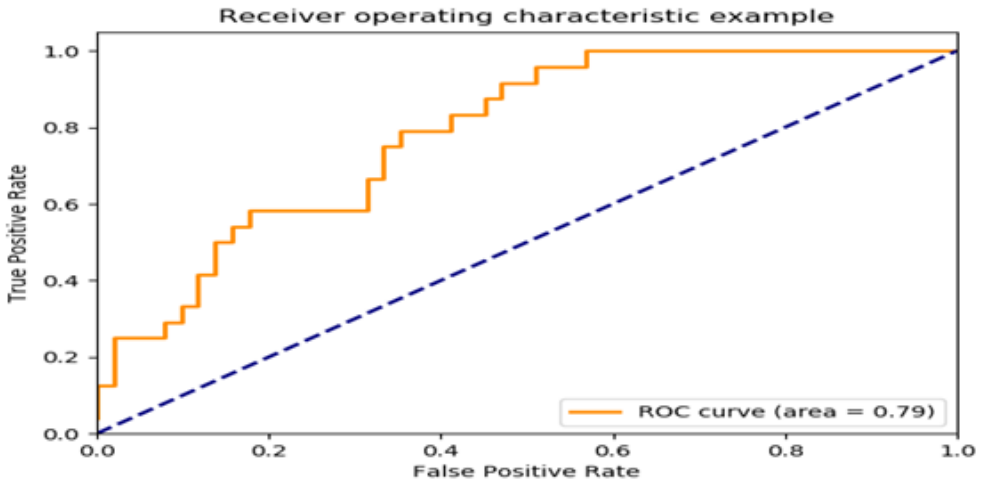

FIGURE 4.1: ROC Curve

TABLE 4.1: Detailed Accuracy by Class For Naive Bayes Algorithm

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.738	0.003	0.992	0.738	0.846	0.959	Normal
0.584	0.124	0.045	0.584	0.084	0.858	Backdoor
0.275	0.031	0.092	0.275	0.138	0.884	Analysis
0.227	0.031	0.461	0.227	0.304	0.891	Fuzzers
0.976	0.223	0.028	0.976	0.054	0.943	Shellcode
0.002	0.014	0.011	0.002	0.004	0.785	Reconnaissance
0.266	0.007	0.895	0.266	0.410	0.771	Exploits
0.006	0.002	0.209	0.006	0.011	0.794	Dos
0.208	0.020	0.008	0.208	0.015	0.880	Worms
0.975	0.009	0.970	0.975	0.972	0.988	Generic

=== Confusion Matrix ===

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

Table 4.2: Confusion Matrix Result For Naive Bayes Algorithm

a	B	c	D	E	F	g	H	I	j	Classified as
41328	1847	2263	3181	6266	369	202	109	126	309	a=Normal
0	1019	56	10	593	4	8	3	25	28	b=Backdoor
12	988	550	2	407	3	4	2	4	28	c=Analysis
128	1982	88	4124	10145	1227	129	12	33	316	d=Fuzzers
0	0	0	23	1106	0	0	0	0	4	e=Shellcode
1	1265	63	151	8858	26	20	4	15	88	f=Reconnaissance
190	8646	2317	1174	8670	436	8891	117	2699	253	g=Exploits
13	6572	596	217	3422	163	563	69	452	197	h=Dos
1	0	2	7	92	1	0	0	27	0	i=Worms
2	237	36	48	317	40	119	14	177	39010	j=Generic

4.4 Decision Tree Algorithm

4.4.1 Performance Evaluation

The efficiency of any machine learning is determined using measures such as true positive rate, false positive rate, true negative rate and false negative rate. Sensitivity and specificity are used to explain clinical diagnostic test and to estimate how good the test was. Training and evaluating statistical performance on same data yields over optimistic result. Cross validation improves performance accuracy. Cross validation is a statistical method that compares machine learning schemes by dividing data into train and test set. The train set is used to train the data and test set is used to validate the model. In k-fold cross validation, data is partitioned into k equal size folds. The k iterations are trained and validated such that within each iteration, different fold of the data is held out for validation and remaining k-1 fold is used for learning. To calculate the accuracy the Evaluation classes of weka is used in which we collect every group of predictions for current model in a Fast Vector. Finally, the model builder builds a model and returns summary of training testing pair using the Evaluation object of summary string method, class detail string and matrix string methods.

Time taken to build model: 44.8 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.74 seconds

=== Summary ===

Correctly Classified Instances 158198 90.2231 %

Incorrectly Classified Instances 17143 9.7769 %

Kappa statistic:0.8753

Mean absolute error: 0.0258

Root mean squared error:0.1136

Relative absolute error: 16.3391 %

Total Number of Instances:175341

Table 4.3: Detailed Accuracy by Class For Decision Tree Algorithm

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1.00	0.000	1.000	1.000	1.000	1.000	Normal
0.313	0.001	0.736	0.313	0.439	0.961	Backdoor
0.286	0.000	0.878	0.286	0.431	0.960	Analysis
0.908	0.003	0.976	0.908	0.941	0.993	Fuzzers
0.914	0.001	0.889	0.914	0.902	0.999	Shellcode
0.798	0.002	0.962	0.798	0.872	0.988	Reconnaissance
0.909	0.080	0.727	0.909	0.808	0.977	Exploits
0.419	0.028	0.532	0.419	0.469	0.952	Dos
0.700	0.000	0.929	0.700	0.798	0.999	Worms
0.990	0.000	0.998	0.990	0.994	1.000	Generic

=== Confusion Matrix ===

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

Table 4.4: Confusion Matrix Result For Decision Tree Algorithm

A	B	c	d	e	f	g	h	I	j	Classified as
5600	0	0	0	0	0	0	0	0	0	a=Normal
0	546	0	7	1	27	766	394	1	4	b=Backdoor
0	111	571	6	0	1	918	392	0	1	c=Analysis
0	20	30	16508	41	22	1109	447	0	7	d=Fuzzers
0	1	0	58	1036	9	23	6	0	0	e=Shellcode
0	29	8	26	0	8369	1513	543	1	1	f=Reconnaissance
0	16	22	207	46	221	30358	2486	3	34	g=Exploits
0	14	18	73	35	51	6923	5138	0	12	h=Dos
0	1	0	5	0	1	27	5	91	0	i=Worms
0	4	1	28	6	1	126	251	2	39581	j=Generic

4.5 Random Forest Algorithm

4.5.1 Performance Evaluation

The efficiency of any machine learning is determined using measures such as true positive rate, false positive rate, true negative rate and false negative rate. Sensitivity and specificity are used to explain clinical diagnostic test and to estimate how good the test was. Training and evaluating statistical performance on same data yields over optimistic result. Cross validation improves performance accuracy. Cross validation is a statistical method that compares machine learning schemes by dividing data into train and test set. The train set is used to train the data and test set is used to validate the model. In k-fold cross validation, data is partitioned into k equal size folds. The k iterations are trained and validated such that within each iteration, different fold of the data is held out for validation and remaining k-1 fold is used for learning.

To calculate the accuracy the Evaluation classes of weka is used in which we collect every group of predictions for current model in a Fast Vector. Finally, the model builder builds a model and returns summary of training testing pair using the Evaluation object of summary string method, class detail string and matrix string methods.

=== Evaluation on training set ===

Time taken to test model on training data: 3.25 seconds

=== Summary ===

Correctly Classified Instances: 146307 83.4414 %

Incorrectly Classified Instances: 29034 16.5586 %

Kappa statistic: 0.7867

Mean absolute error: 0.1614

Root mean squared error: 0.2743

Relative absolute error: 102.1019 %

Total Number of Instances: 175341

Table 4.5: Detailed Accuracy by Class For Random Forest Algorithm

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	Normal
0.006	0.000	0.440	0.006	0.012	0.715	Backdoor
0.077	0.000	0.962	0.077	0.142	0.816	Analysis
0.802	0.029	0.759	0.802	0.780	0.914	Fuzzers
0.000	0.000	0.004	0.000	0.120	0.943	Shellcode
0.570	0.017	0.679	0.570	0.620	0.931	Reconnaissance
0.911	0.151	0.586	0.911	0.713	0.893	Exploits
0.003	0.000	0.646	0.003	0.007	0.847	Dos
0.000	0.000	0.432	0.000	0.002	0.940	Worms
0.978	0.001	0.998	0.978	0.988	0.997	Generic

=== Confusion Matrix ===

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

Table 4.6: Confusion Matrix Result For Random Forest Algorithm

a	B	C	d	e	f	g	H	i	J	Classified as
56000	0	0	0	0	0	0	0	0	0	a=Normal
0	11	0	62	0	101	1572	0	0	0	b=Backdoor
0	0	153	0	0	0	1847	0	0	0	c=Analysis
0	0	6	14592	0	1252	2308	3	0	23	d=Fuzzers
0	0	0	437	0	696	0	0	0	0	e=Shellcode
0	0	0	1080	0	5982	3416	0	0	13	f=Reconnaissance
0	10	0	2413	0	518	30408	20	0	24	g=Exploits
0	4	0	469	0	214	11504	42	0	31	h=Dos
0	0	0	11	0	6	113	0	0	0	i=Worms
0	0	0	156	0	38	687	0	0	39119	j=Generic

4.6 Support Vector Machine Algorithm

4.6.1 Performance Evaluation

The efficiency of any machine learning is determined using measures such as true positive rate, false positive rate, true negative rate and false negative rate. Sensitivity and specificity are used to explain clinical diagnostic test and to estimate how good the test was. Training and evaluating statistical performance on same

data yields over optimistic result. Cross validation improves performance accuracy. Cross validation is a statistical method that compares machine learning schemes by dividing data into train and test set. The train set is used to train the data and test set is used to validate the model. In k-fold cross validation, data is partitioned into k equal size folds. The k iterations are trained and validated such that within each iteration, different fold of the data is held out for validation and remaining k-1 fold is used for learning.

To calculate the accuracy the Evaluation classes of weka is used in which we collect every group of predictions for current model in a Fast Vector. Finally, the model builder builds a model and returns summary of training testing pair using the Evaluation object of summary string method, class detail string and matrix string methods.

Time is taken to build the model: 141.99 seconds

=== Evaluation on training set ===

Time is taken to test the model on training data: 8.31 seconds

=== Summary ===

Correctly Classified Instances: 175336 99.9971 %

Incorrectly Classified Instances: 5 0.0029 %

Kappa statistic: 1

Mean absolute error: 0.0113

Root mean squared error: 0.0517

Relative absolute error: 7.1685 %

Total Number of Instances: 175341

Table 4.7: Detailed Accuracy by Class For Support Vector Machine Algorithm

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.989	0.001	1.000	1.000	1.000	0.998	Normal
1.000	0.000	1.000	1.000	1.000	1.000	Backdoor
1.000	0.000	1.000	1.000	1.000	1.000	Analysis
1.000	0.000	1.000	1.000	1.000	1.000	Fuzzers
1.000	0.000	1.000	1.000	1.000	1.000	Shellcode
1.000	0.000	1.000	1.000	1.000	1.000	Reconnaissance
1.000	0.000	1.000	1.000	1.000	1.000	Exploits
1.000	0.000	1.000	1.000	0.994	1.000	Dos
1.000	0.000	1.000	1.000	1.000	1.000	Worms
1.000	0.002	1.000	1.000	1.000	1.000	Generic

=== Confusion Matrix ===

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

Table 4.8: Confusion Matrix Result For Support Vector Machine Algorithm

A	B	c	D	e	f	g	h	i	j	Classified as
56000	0	0	0	0	0	0	0	0	0	a=Normal
0	1746	0	0	0	0	0	0	0	0	b=Backdoor
0	0	2000	0	0	0	0	0	0	0	c=Analysis
0	0	0	18182	0	0	2	0	0	0	d=Fuzzers
0	0	0	0	1133	0	0	0	0	0	e=Shellcode
0	0	0	0	0	10490	0	1	0	0	f=Reconnaissance
0	0	0	0	0	0	33392	1	0	0	g=Exploits
0	0	0	0	0	0	1	12263	0	0	h=Dos
0	0	0	0	0	0	0	0	130	0	i=Worms
0	0	0	0	0	0	0	0	0	40000	j=Generic

Comparison of accuracy of the training data with different machine learning algorithms:

Table 4.9: Accuracy for training data

Machine learning algorithm	Correctly classified (%)	Incorrectly classified (%)
Naïve Bayes	54.836	45.164
Decision Tree	90.2231	9.7769
Random Forest	83.4414	16.5586
Support Vector Machine	99.9971	0.0029

4.7 Validate the Model

After the data is trained with different machine learning algorithms (Naive Bayes, Decision Tree, Random Forest, and Support Vector Machine)in WEKA we get a model for each algorithm. The next phase is to validate the model with fresh data(test data) with different machine learning algorithms in WEKA. finally, we select the best-performed algorithm among them to validate the model.

4.7.1 Naïve Bayes Algorithm

The following are results for Naive Bayes algorithm in WEKA after test the data:

Time taken to build model: 1.22 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 7.51 seconds

=== Summary ===

Correctly Classified Instances: 36907 44.827 %

Incorrectly Classified Instances: 45425 55.173 %

Kappa statistic: 0.3498

Mean absolute error: 0.1104

Root mean squared error: 0.3269

Relative absolute error: 72.1678 %

Total Number of Instances: 82332

The table below shows the accuracy result for each class after test the data with Naive Bayes algorithm.

Table 4.10: Naive Bayes Accuracy by Class For Naive Bayes Algorithm

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.479	0.004	0.990	0.479	0.646	0.882	Normal
0.806	0.165	0.034	0.806	0.065	0.870	Backdoor
0.084	0.060	0.012	0.084	0.020	0.844	Analysis
0.219	0.053	0.248	0.219	0.232	0.814	Fuzzers
0.987	0.230	0.019	0.987	0.038	0.928	Shellcode
0.002	0.016	0.005	0.002	0.003	0.766	Reconnaissance
0.329	0.010	0.842	0.329	0.473	0.766	Exploits
0.008	0.001	0.216	0.008	0.015	0.729	DoS
0.205	0.018	0.006	0.205	0.012	0.867	Worms
0.702	0.008	0.961	0.702	0.812	0.973	Generic

=== Confusion Matrix ===

The table below shows the result for confusion matrix after test the data with Naive Bayes algorithm in WEKA. From the table the diagonal values are correctly classified instances and from the diagonal below and above values are incorrectly classified instances.

Table 4.11: Confusion Matrix For Naive Bayes Algorithm

a	b	c	d	e	f	g	h	i	j	<– classified as
17720	1824	3742	3362	9202	546	279	76	63	186	a = Normal
0	470	15	2	52	0	2	0	3	39	b = Backdoor
2	524	57	2	54	0	0	0	0	38	c = Analysis
64	1320	4	1327	2757	405	36	1	14	134	d = Fuzzers
0	0	0	5	373	0	0	0	0	0	e = Shellcode
0	258	7	32	3176	7	7	0	5	4	f=Reconnaissance
89	2585	807	460	2239	191	3662	32	979	88	g = Exploits
5	2234	253	106	886	75	257	32	196	45	h = DoS
0	1	0	2	32	0	0	0	9	0	i = Worms
15	4740	38	58	420	52	108	7	183	13250	j = Generic

The ROC Curve

The ROC Cure is a plot of true positive rate against false positive rate for the trained classifier that is selected at difference points in [0, 1].

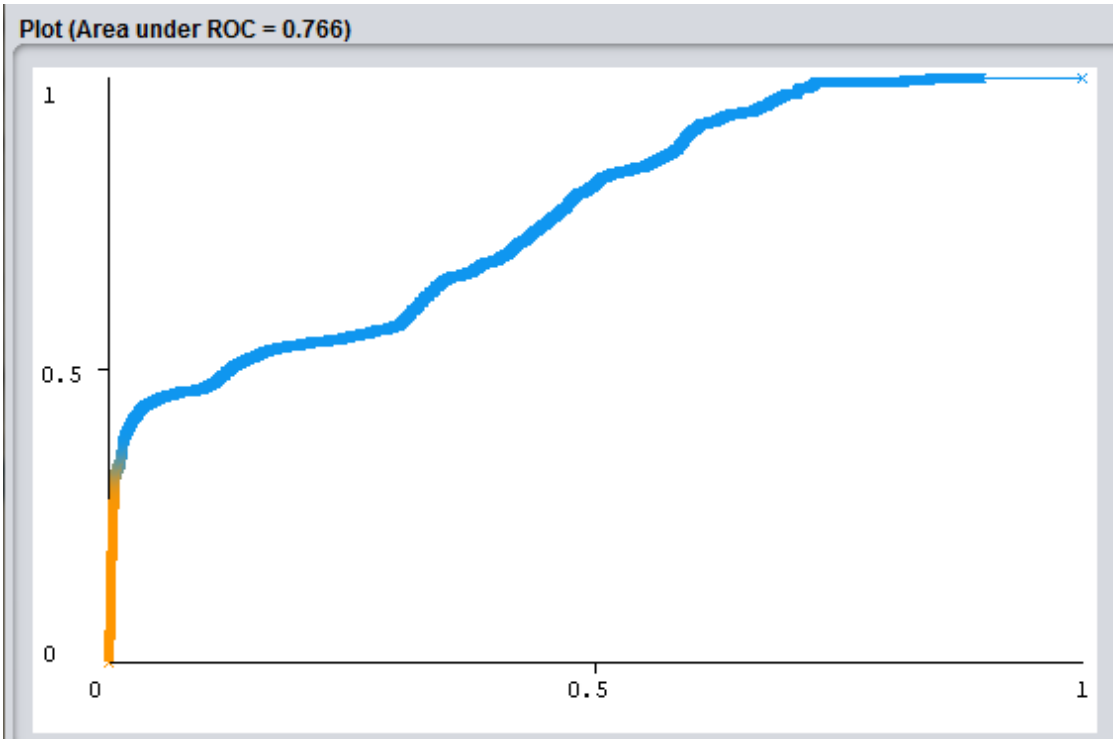

FIGURE 4.2: ROC Curve For Naive Bayes Algorithm

4.7.2 Decision Tree Algorithm

The following are the result for Naive Bayes algorithm in WEKA after the test the data:

Time taken to build model: 43.79 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.79 seconds

=== Summary ===

Correctly Classified Instances: 71153 88.422 %

Incorrectly Classified Instances: 11179 11.578 %

Kappa statistic: 0.8114

Mean absolute error: 0.0286

Root mean squared error: 0.1456

Relative absolute error: 18.7349 %

Total Number of Instances: 82332

The table below shows the accuracy result for each class after test the data with Decision Tree algorithm.

Table 4.12: Detailed Accuracy by Class For Decision Tree Algorithm

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	Normal
0.161	0.060	0.019	0.161	0.034	0.894	Backdoor
0.007	0.002	0.030	0.007	0.012	0.890	Analysis
0.675	0.007	0.891	0.675	0.768	0.917	Fuzzers
0.701	0.004	0.423	0.701	0.528	0.888	Shellcode
0.805	0.003	0.917	0.805	0.857	0.922	Reconnaissance
0.731	0.061	0.653	0.731	0.690	0.855	Exploits
0.127	0.007	0.477	0.127	0.201	0.641	Dos
0.591	0.000	0.419	0.591	0.491	0.917	Worms
0.964	0.001	0.997	0.964	0.980	0.984	Generic

=== Confusion Matrix ===

The following table shows the result for confusion matrix after test the data with Decision tree algorithm in WEKA.From the table the diagonal values are correctly classified instances and from the diagonal below and above values are incorrectly classified instances.

TABLE 4.13: Confusion Matrix Result For Decision Tree Algorithm

A	B	c	d	E	f	G	h	I	j	Classified as
37000	0	0	0	0	0	0	0	0	0	a=Normal
0	94	5	11	6	1	456	10	0	0	b=Backdoor
0	178	5	1	3	0	487	3	0	0	c=Analysis
0	331	23	4093	178	20	1284	90	11	32	d=Fuzzers
0	14	0	28	265	5	55	10	1	0	e=Shellcode
0	293	13	27	23	2814	304	20	2	0	f=Reconnaissance
0	1984	70	276	95	198	8135	339	15	20	g=Exploits
0	2096	46	102	39	31	1249	521	2	3	h=Dos
0	2	0	1	0	0	12	2	26	1	i=Worms
0	25	2	54	17	0	470	98	5	18200	j=Generic

The ROC Curve

The ROC Cure is a plot of true positive rate against false positive rate for the trained classifier that is selected at difference points in [0, 1].

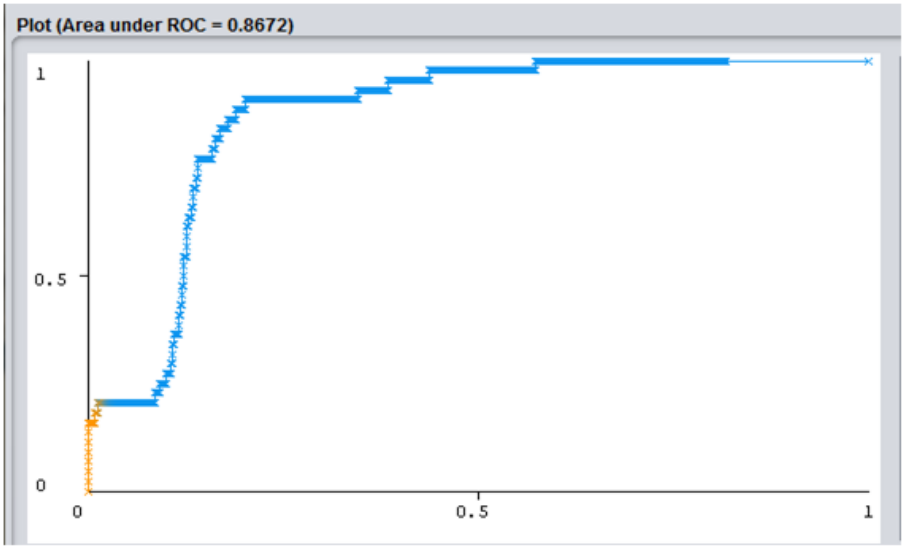

FIGURE 4.3: ROC Curve For Decision Tree Algorithm

4.7.3 Random Forest Algorithm

The following are the result for Random forest algorithm in WEKA after the test the data:

Time taken to build model: 146485.66 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 2.2 seconds

=== Summary ===

Correctly Classified Instances 53446 80.9152 %

Incorrectly Classified Instances 28886 19.0848 %

Kappa statistic: 0.5295

Mean absolute error: 0.1651

Root mean squared error: 0.2805

Relative absolute error: 107.9381 %

Total Number of Instances: 82332

The table below shows the accuracy result for each class after test the data with Decision tree algorithm.

Table 4.14: Detailed Accuracy by Class For Random Forest Algorithm

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	Normal
0.000	0.000	0.000	0.000	0.000	0.726	Backdoor
0.000	0.000	0.000	0.000	0.000	0.737	Analysis
0.723	0.266	0.178	0.723	0.286	0.784	Fuzzers
0.000	0.000	0.003	0.000	0.005	0.947	Shellcode
0.651	0.011	0.716	0.651	0.682	0.959	Reconnaissance
0.878	0.108	0.559	0.878	0.683	0.906	Exploits
0.004	0.000	0.789	0.004	0.007	0.854	DoS
0.000	0.000	0.001	0.000	0.002	0.892	Worms
0.000	0.000	0.009	0.000	0.004	0.961	Generic

=== Confusion Matrix ===

The following table shows the result for confusion matrix after test the data with Random forest algorithm in WEKA.From the table the diagonal values are correctly classified instances and from the diagonal below and above values are incorrectly classified instances.

TABLE 4.15: Confusion Matrix Result For Random Forest Algorithm

a	b	c	D	e	f	G	h	i	j	classified as
37000	0	0	0	0	0	0	0	0	0	a = Normal
0	0	0	16	0	26	541	0	0	0	b = Backdoor
0	0	0	0	0	0	677	0	0	0	c = Analysis
0	0	0	4385	0	168	1509	0	0	0	d = Fuzzers
0	0	0	138	0	239	1	0	0	0	e = Shellcode
0	0	0	420	0	2275	801	0	0	0	f = Reconnaissance
0	2	16	1031	0	309	9771	3	0	0	g = Exploits
0	3	0	2	61	0	113	3697 15	0	0	h = DoS
0	0	0	9	0	1	34	0	0	0	i = Worms
0	0	0	18386	0	46	438	1	0	0	j = Generic

The ROC Curve

The ROC Cure is a plot of true positive rate against false positive rate for the trained classifier that is selected at difference points in [0, 1].

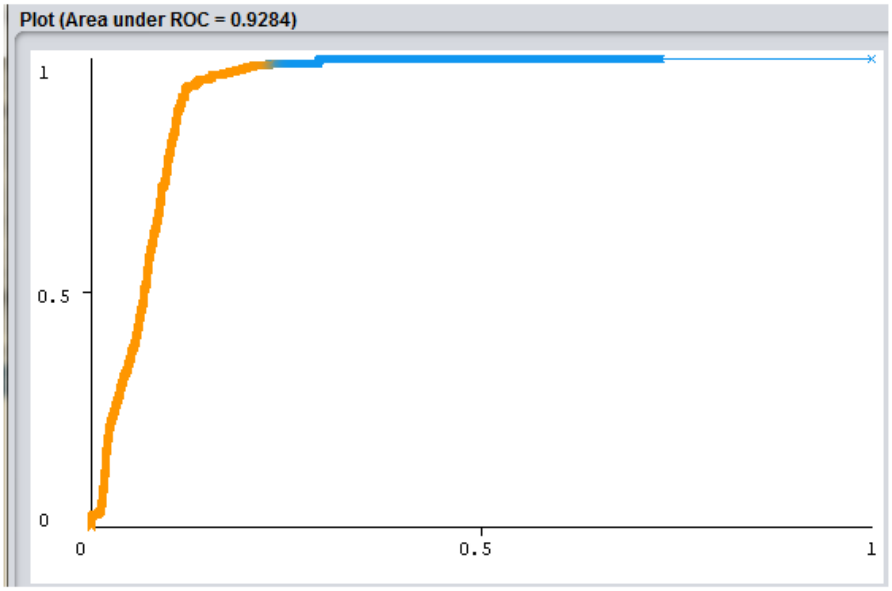

FIGURE 4.4: ROC Curve For Random Forest Algorithm

4.7.4 Support Vector Machine Algorithm

The following are the result for Support Vector Machine algorithm in WEKA after the test the data:

Time is taken to build the model: 141.99 seconds

=== Evaluation on training set ===

Time is taken to test the model on training data: 8.31 seconds

=== Summary ===

Correctly Classified Instances: 175330 98.2971 %

Incorrectly Classified Instances: 11 1.7029%

Kappa statistic: 1

Mean absolute error: 0.0113

Root mean squared error: 0.0517

Relative absolute error: 7.1685 %

Total Number of Instances: 175341

The following table shows the accuracy result for each class after test the data with Support Vector Machine algorithm.

Table 4.16: Detailed Accuracy by Class For Support Vector Machine Algorithm

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.998	0.000	0.995	1.000	1.000	1.000	Normal
1.000	0.001	1.000	1.000	0.998	0.974	Backdoor
1.000	0.000	1.000	0.988	1.000	1.000	Analysis
0.999	0.000	1.000	1.000	0.952	0.995	Fuzzers
1.000	0.001	1.000	0.992	1.000	1.000	Shellcode
1.000	0.000	0.997	1.000	1.000	0.976	Reconnaissance
0.997	0.000	1.000	0.943	0.997	1.000	Exploits
1.000	0.000	1.000	1.000	1.000	1.000	Dos
1.000	0.002	0.996	0.987	1.000	1.000	Worms
0.996	0.000	1.000	1.000	0.977	1.000	Generic

=== Confusion Matrix ===

The following table shows the result for confusion matrix after train the data with Naive Bayes algorithm in WEKA.From the table the diagonal values are correctly classified instances and from the diagonal below and above values are incorrectly classified instances.

TABLE 4.17: Confusion Matrix Result For Support Vector Machine Algorithm

a	B	c	D	E	f	g	h	i	j	Classified as
56000	0	3	0	0	0	0	2	0	3	a=Normal
0	1746	0	0	0	0	0	0	0	0	b=Backdoor
0	0	2000	0	0	0	0	0	0	0	c=Analysis
0	0	0	18182	0	0	2	0	0	0	d=Fuzzers
0	0	0	0	1133	0	0	0	1	0	e=Shellcode
5	0	0	0	0	10490	0	1	0	0	f=Reconnaissance
0	0	0	1	0	0	33392	1	0	0	g=Exploits
0	0	0	0	0	0	1	12263	0	0	h=Dos
0	0	0	0	0	1	0	0	130	0	i=Worms
0	0	0	0	0	0	0	0	0	40000	j=Generic

The ROC Curve

The ROC Cure is a plot of true positive rate against false positive rate for the trained classifier that is selected at difference points in [0, 1].

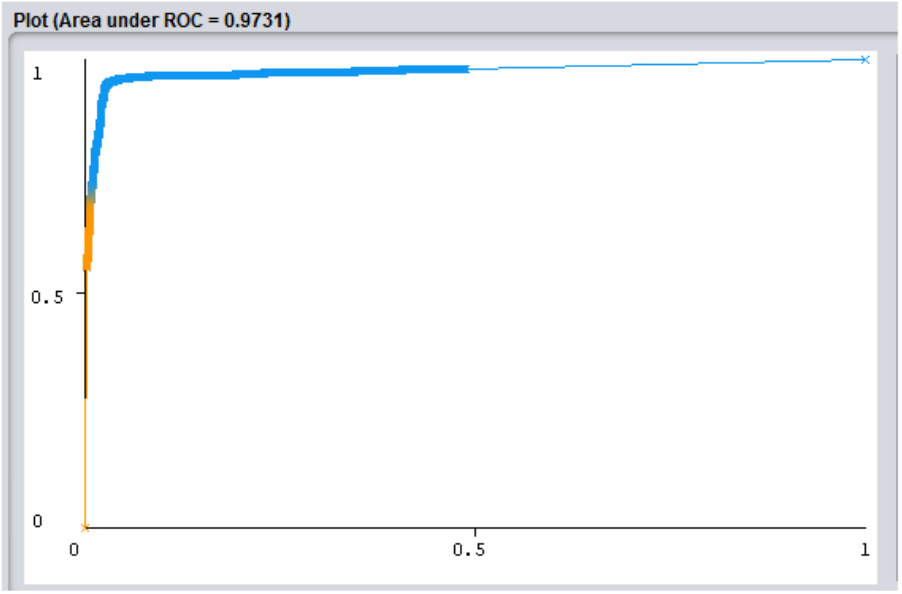

FIGURE 4.5: ROC Curve For Support Vector Machine Algorithm

Comparison of accuracy of the testing data with different machine learning algorithm:

TABLE 4.18: Accuracy for test data

Machine learning algorithm	Correctly classified (%)	Incorrectly classified (%)
Naïve Bayes	44.827	55.173
Decision tress	88.422	11.578
Random forest	80.9152	19.0848
Support Vector Machine	98.2971	1.7029

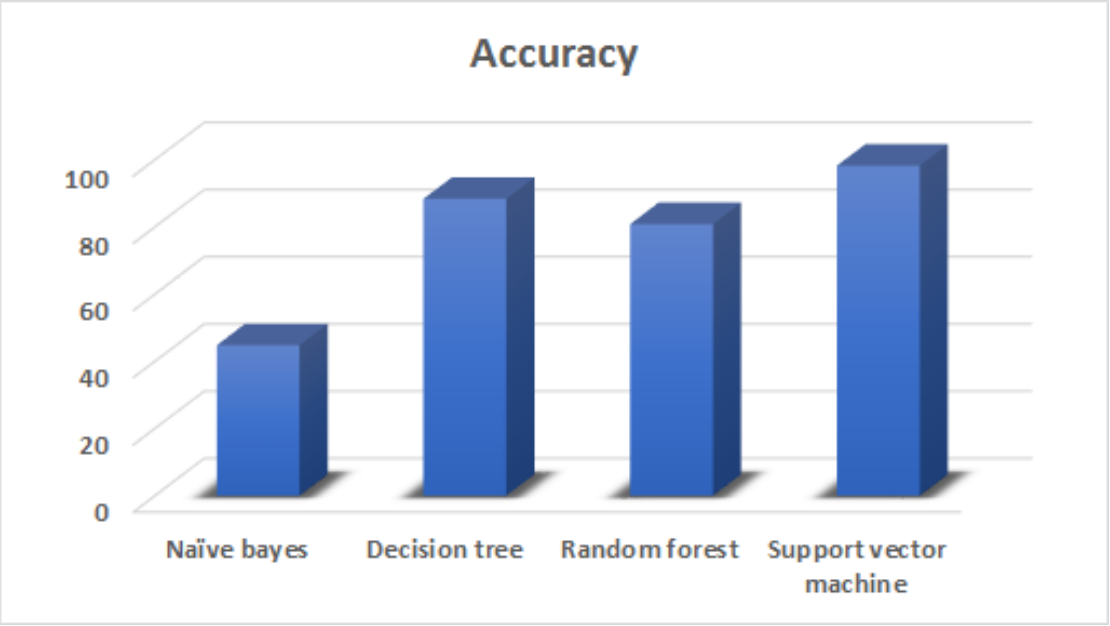

FIGURE 4.6: Accuracy result for test data with each Algorithm

From the below table, we compare our proposed model to some existing proposed model based on different requirements, and the proposed model by cascading Random forest and SVM is best performed among others.

TABLE 4.19: Compare our proposed model to some existing proposed model

Machine Learning Algorithm	**Selected attributes**	**Number of attacks in the model**	**Accuracy**
Cascading Random Forest and Naïve Bayes	20	9	44,827%
Cascading Random Forest and Decision Tree	20	9	88.422%
Cascading SVM and Decision Tree	21	4	94.2%
Cascading SVM and Decision Tree	30	4	95.3%
Cascading SVM and Decision Tree	36	4	96.2%
Cascading Random Forest and SVM	**20**	**9**	**98.2971%**

Chapter 5

Conclusion and Recommendation

5.1 Conclusions

IoT is anew emerging technology now a day billions of devices are integrated that facilitate human life easy and safe but IoT systems are vulnerable to a network attack, physical attacks, software attacks, and privacy leakage. To handle this problem Intrusion Detection System is developed to make IoT system safe to users.

In this paper, we used a random forest algorithm for feature selection mechanisms and four(4) different supervised machine learning algorithms (Naive Bayes, Decision tree, Random forest, and Support Vector Machine)for classification of attacks in IoT network. The experimental result shows that by cascading(Random forest for feature selection and Naive Bayes for classification of attacks)is 44.827% accuracy, cascading(Random forest for feature selection and Decision tree for classification of attacks) is 88.422% accuracy, cascading (Random forest for feature selection and Random forest for classification of attacks)is 80.9152% accuracy and cascading(Random forest for feature selection and Support Vector Machine for classification of attacks)is 98.2971% accuracy. From the experimental result cascading (Random forest and Support Vector Machine) is effectively detect the intruders in the IoT network with different performance metrics.

5.2 Recommendations

Intrusion detection in IoT is much of interest of researchers.he-nice, recent researchers focus on how to improve an algorithm to improve the accuracy of detection mechanisms in IoT.but, in this research hybrid Intrusion Detection System by cascading random forest and Support Vector Machine algorithm is developed to overcome the security problems in IoT network. In the future:

1. The researcher considers newborn attack in IoT network.

2. In this research, we select four(4) machine learning algorithm for classification of attacks in IoT network for the future the researcher add other machine learning algorithm for comparison of performance.

3. In this research, we used a machine learning approach to construct the model for the future the researcher select other approaches like a deep learning approach.

Bibliography

[1] E. Borgia, "The Internet of Things vision: Key features, applications, and open issues," Computer Communications, 2014, vol. 54, pp. 1–31. doi: 10.1016/j.comcom.2014.09.008

[2] J. Vacca, "Computer and Information Security Handbooks Morganu Kaufmann," Amsterdam, 2013.

[3] R. Mitchell and Ing-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," ACM Computing Surveys (CSUR), 2014, vol. 46, No. 4, Article 55 (April 2014), doi: 10.1145/2542049

[4] R. Mitchell and Ing-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," ACM Computing Surveys (CSUR), 2014, vol. 46, No. 4, 29 pages.

[5] Symantec, "Internet security threat report," Technical report, vol. 22, April 2017.

[6] E. Borgia,"The Internet of Things vision: Key features, application and open issues in technology and standardization," Computer Communications, 2014, vol. 54, doi: 10.1016/j.comcom.2014.09.008.

[7] M.U. Farooq, M. Waseem, A. Khairi and S. Mazhar, "A Critical Analysis on the Security Concerns of Internet of Things (IoT)," International Journal of Computer Applications, 2015, vol. 111, No. 7, pp. 1-6.

[8] "IEEE Standard for Local and metropolitan area networks", part 154, 2011.

[9] K. Zhao and L. Ge, "A survey on the internet of things security," in Int'l Conf. on Computational Intelligence and Security (CIS), 663-667, 2013.

[10] S. Sicari, A. Rizzardi, L.A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of things: The road ahead," Computer Networks, vol. 76, pp. 146-164, Jan. 2015.

[11] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," Computer Networks, vol. 56, 3594-3608, 2012.

[12] M. Leo, F. Battisti, M. Carli, and A. Neri, "A federated architecture approach for Internet of Things security," in Euro Med Telco Conference (EMTC), 1-5, 2014.

[13] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the internet of things," Ad Hoc Networks, vol. 11, pp. 2661-2674, Nov. 2013.

[14] M. Abomhara and G. M. Koien, "Security and privacy in the Internet of Things: Current status and open issues," in Int'l Conference on Privacy and Security in Mobile Systems (PRISMS), 1-8, 2014.

[15] Q. Wen, X. Dong, and R. Zhang, "Application of dynamic variable cipher security certificate in internet of things," in Int'l Conference on Cloud Computing and Intelligent Systems (CCIS), 1062-1066, 2012.

[16] R. Roman, P. Najera, and J. Lopez, "Securing the internet of things," Computer, vol. 44, 51-58, 2011.

[17] M. Farooq, M. Waseem, A. Khairi, and S. Mazhar, "A Critical Analysis on the Security Concerns of Internet of Things (IoT)," Perception, vol. 111, 2015.

[18] Tom M. Mitchell, "Machine Learning," March 1, 1997, pp. 3-7.

[19] A. Smola and S.V.N. Vishwanathan, "Introduction to Machine Learning," Departments of Statistics and Computer Science Purdue University and College of Engineering and Computer Science, 2010, Australian National University.

[20] S.zha.p“Machine learning: A review of classification and combining techniques” pp.2-6 2014, DOI:10.1007/s10462-007-9052-3.

[21] H. Zhang, ”The Optimality of Naive Bayes,” Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004. vol. 2.

[22] R. Caruana and A. Niculescu-Mizil, ”An empirical comparison of supervised learning algorithms,” Proceedings of the 23rd international conference on Machine learning, 2006.

[23] C. Cervantes, D. People, M. Nogueira, and A. Santos, ”Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for the Internet of Things,” In 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 606-611, 2015.

[24] N. Thanigaivelan, E. Nigussie, R. Kanth, S. Virtanen, and J. Isoaho, ”Distributed internal anomaly detection system for Internet-of-Things,” In 2016 Proceedings of the 13th IEEE Annual Consumer Communications Networking Conference (CCNC), pp. 319-320, 2016.

[25] S. Prachi, ”ML-IDS Machine Learning Approach to detect wormhole Attacks in the Internet of Things,” Intelligent System Conference, 2017.

[26] K. John and B. James ”Hybrid Intrusion Detection System in the Internet of Things using Support Vector Machine and Decision Tree,” International Journal of Computer Applications, 2018.

[27] T.A. Mohamed, T. Otsuka and T. Ito, ”Towards Machine Learning-Based IoT Intrusion Detection Service Conference,” The 31st International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems Cite this publication 2018.

[28] S. Chawla and G. Thamilarasu, ”Security as a Service: Real-time Intrusion Detection in the Internet of Things,” The Fifth Conference on Cybersecurity Symposium 2018.

[29] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 network data set)," 2015 Military Communications and Information Systems Conference (MilCIS), 2015.

[30] T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), 2017.

[31] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 network data set)," Military Communications and Information Systems Conference (MilCIS),2015.

[32] https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo4ys 8:30 AM June 20,2019.

[33] T.A. Mohamed, T. Otsuka and T. Ito, "Towards Machine Learning-Based IoT Intrusion Detection Service Conference," The 31st International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems Cite this publication 2018.

[34] Weka Machine Learning Project, http://www.cs.waikato.ac.nz/ ml/index.html.

[35] E. Frank, Machine Learning With WEKA, University of Waikato, New Zealand.