LIST OF FIGURES

Figure

LIST OF TABLES

Table

TABLE OF CONTENTS

CHAPTER

CHAPTER I

Introduction

1.1 Overview

This section includes the overview of the report.

1.2 Problem Statement

This section includes problem statement

1.3 Motivation

Analyzing research interest and existing work in the field of IoT, security is the most vital issue that drives the researcher towards the field. As architecture and application is going large day by day, security has become the important challenge for IoT. Interest in this field is proportionally increases with the Heterogeneity of this network.

In figure 1.1 shows the existing research interest in IoT Attack Detection which is increasing day by day for last few years whereas for detecting attack concept of Adaptive Firewall is not so common and used term in this field. This drives us

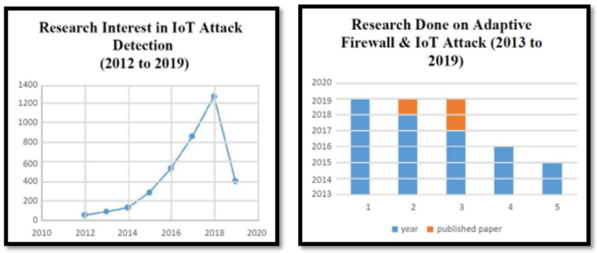


Figure 1.1: Research Interest in Field of IoT

motivated to design adaptive firewall for attack detection and to block illegitimate traffic on IoT Network Model.

1.4 Objective

IoT network model and devices are vulnerable to different kind of attacks. These attacks may vary to different category, so have different approach to detect and block them. The goal of this research is to study and identify potential IoT security attacks, detect and mitigate them by using Adaptive firewall concept. Additionally, machine learning should be considered for classifying attacks and identifying attacks. Specific goals of this thesis that should be mentioned:

- Analyze network traffic to detect the malicious ones that tries to hamper the network.
- Find the characteristics of perception layer's attack to identify specific attacks.
- Extract features from the generated traffic datasets to train machine learning classifiers and apply them to recognize attacks.Propose a centralized attack detection model.
- Design a rule based and ANN based FIS to help SDN controller to evaluate specific attack probability and block the suspicious ones.
- Maintaining the performance of the network.

This proposed model also answers the following questions:

1. What are the major challenges that have guided security in IoT?
2. What is the best attack detection way for IoT network model?
3. Is there any generalized approach for detecting different layer attack?
4. What is the best way to detect a single layer attack?

1.5 Assumptions & Limitations

Though a centralized and efficient model has been proposed to detect attacks on IoT network, it has limitations on which further studies should be done:

- No approach has been mentioned for network and application layer security.

- No real time data has been used for the traffic analysis.
- Here feature Extraction and Selection method has been analyzed but no implementation has been shown.
- No comparison among different IDS model has been analyzed so can't be declared it as the optimal way.
- No performance measure of used Classifier is evaluated here.

1.6 Research Outline

Rest of the report is structured as follows: In **Chapter II** a literature study on related work is given including explanations for the most important terms used in this thesis-basic concept and architecture of IoT Network Model, Attacks on IoT, Concept and architecture of SDN, different model of IDS, Concept of Firewall has been discussed through this chapter. **Chapter III** introduces system model including system architecture, algorithm and flowchart of working procedure of entire system model. **Chapter IV** explains the details of traffic analysis techniques, Feature Extraction and Selection mechanism and tools for this mechanism and reasoning how these mechanisms work for our model. **Chapter V** discusses about the simulation and model performance, to analysis result of the model it describes the basic mechanism of attack detection like Fuzzification, NSL KDD dataset, FIS, Defuzzification,Simulation and confusion matrix. Lastly in **Chapter VI** future work and conclusion is mentioned.

CHAPTER II

Literature Review

2.1 Related Work

Several studies investigated diabetes data and constructed models to predict diabetes. Reference [1] discussed several classifiers, Support Vector Machine, Naive Bayes and Decision as a base classifiers for AdaBoost calculation for accuracy calculation. The accuracy received for AdaBoost calculation with base classifier is 80.72%.

2.2 Machine Learning Types

2.3 Supervised Machine Learning Classifiers

In this step, we will describe some supervised machine learning classifiers named Logistic Regression, k-nearest neighbors, Support Vector Machine, Decision Tree, Gaussian Naive Bayes, Random Forest, Gradient Boosting and Linear Discriminant Analysis.

2.3.1 Logistic Regression (LR)

Logistic Regression (LR) is a supervised machine learning data classification algorithm that mines real-valued features from the input, multiplies each of them by a weight, adds them, and transfers the sum through a sigmoid function to produce a probability. A threshold is used to finalize a decision [2]. A solution for classification of our data set is LR which Instead of fitting a straight line or hyperplane uses the logistic function to squeeze the output of a linear equation between 0 and 1. The logistic function is defined as:

$$Logistic(\eta) = 1/(1 + exp(-\eta))$$

As η goes from $-\infty$ to ∞, logistic (η) goes from 0 to 1, a “squashing function”. In our study, we used a maximum 4000 iterations to converge the output.

2.3.2 k-nearest neighbors (KNN)

(KNN) is a non-parametric process we used for diabetic data classification. In KNN a data is classified by a majority vote of its neighbors, with the data being allotted to the class most mutual amongst its K nearest neighbors estimated by a distance function. If K = 1, then the data is simply allotted to the class of its nearest neighbor. KNN algorithm is as below [3]:

Algorithm 1 KNN

```
Let $m$ be the number of training data samples. Let $p$ be an unknown point that needs to be classified
Storing the training samples in an array of data points $arr[]$. Each element of this array denotes a tuple $(x, y)$.
for $i = 0$ to $m$ do
    Calculating distance $d(arr[i], p)$
end for
Making set $S$ of $K$ smallest distances achieved. Each of these distances resembles an already classified data point
Returning the majority label among $S$
```

2.3.3 Support Vector Machine (SVM)

A Support Vector Machine is a discriminative classifier well-defined by a separating hyperplane. In other words, specified labelled training data, the algorithm generates an optimal hyperplane which classifies the new data point. In two dimensional space, this hyperplane is a line separating a plane in two parts where each class lay in either side. Along with linear data it also classifies the non-linear data using kernel trick. Hyperplane can be written as the set of points $\vec{x}$ satisfying:

$$\vec{w}.\vec{x} - b = 0$$

The parameter $\frac{b}{\|\vec{w}\|}$ defines the offset of the hyperplane from the origin along the vector $\vec{w}$ which needs to be maximize. SVM uses ”regularization parameter” which controls the trade-off between experimental error and complexity of the assumption space used [4].

2.3.4 Decision Tree (DT)

A DT is a classifier that recursively performs partition of the instance space. The decision tree contains nodes that form a tree, a node called “root” that has no incoming edges is the starting point of the tree. All other nodes have one incoming edge. The leaf nodes are known as decision nodes. The child node is nominated by computing Information Gain (IG).

Information Gain = Entropy(parent) - [weights average] * Entropy(children)

Entropy(Ci) = -P(xi) log P(xi), where P(xi) is the probability of child node i.

Node with the highest IG will be the parent for next level. This process is continued until it gets a leaf node and completed decision tree.

The algorithm for generating a decision tree is as below [5]:

Algorithm 2 DT

```
1: Create (T)
2: Calculate frequencies (Ci, T)
3: If all instances belong to the same class, returning leaf
4: for every attribute a test is set for splitting criteria. An attribute that satisfies
   the test is test node K
5: Repeating Create (Ti) on each partition Ti.Adding those nodes as children of
   node K
```

2.3.5 Gaussian Naive Bayes (GNB)

The GNB classifier is a probability distribution function having the effect of associating neural activation to the means and variances of activation in various impulse conditions. The production of the classifier is a condition-label [6]. The classifier creates hypothesis that the classes have Gaussian normal distributions.

The z-score distance between the inputted point and each class-mean is estimated for each data point, namely the distance from the class mean divided by the standard deviation of that class.

$$Z_{\mathrm{A}} = \frac{(x - \mu_{\mathrm{A}})}{\sigma_{\mathrm{A}}}$$

According to the equation for a Gaussian normal distribution, each z-score is then converted into a probability value which is used for observing data point x. The co-variance between dimensions is not modelled by GNB classifier.

2.3.6 Random Forest (RF)

RF is a collective algorithm which was modelled from trees algorithm and Bagging algorithm. It works fine with a data set with a large number of input variables. It is a meta estimator that creates a number of decision tree classifiers on different subsamples of the data set and uses mean value to increase the accuracy of the model and control over-fitting. Suppose training data set is given as: [X1, X2, X3, X4] with labels as [L1, L2, L3, L4] respectively, random forest algorithm may create three decision trees taking input of subset for example, [X1, X3, X4], [X2, X3, X4] and [X1, X2, X4]. Finally, it predicts class based on the majority of votes from each of the decision trees generated. Generally, the more trees in the forest the more robust and reliable the forest is. The random forest classifier works in the same way, the higher the number of trees in the forest gives higher accuracy output [7].

2.3.7 Gradient boosting (GB)

GB includes three components: a loss function that is to be optimized, a weak learner that makes predictions and an additive model which will add weak learners to minimize the loss function.

2.3.8 Linear Discriminant Analysis (LDA)

The goal of an LDA is to convert a feature space (n-dimensional data set) into a smaller dimension k (where k $\leq$ n-1) while saving the class-discriminatory information. 5 general steps for performing a linear discriminant analysis are given below: [8]

1. Computing the d-dimensional mean vectors for different classes.
2. Generating the scatter matrices.

2.4 Research Gap

Analyzing related works in this field an be noticed some shortcoming in the security measurements of IoT network. **Firstly**, there is no centralized detection method is mentioned, every layer has specific detection way method but as IoT is becoming a heterogenous network a centralized model should be proposed in controller which will control the traffic of every subpart of network. **Secondly**, by using KDD Dataset most commonly **DDoS, Probe, U2R, R2L** attack has been detected but with advancing technology intruder can attack the network in many other ways. **Thirdly**, no time efficient optimal way is mentioned to detect attack. **Fourthly**, traditional firewall can't detect any encrypted incoming packet which can be removed by using adaptive firewall concept but still much work has not done yet regarding this problem.

CHAPTER III

System Model

3.1 Proposed Architecture

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. But now with the rapid growth of information technology and connectivity of devices the security of this complex, heterogeneous IoT network has become a challenging issue. And Software-defined networking (SDN) technology is an approach to network management that enables dynamic, programmatically efficient network configuration in order to improve network performance.

As our main purpose is to secure the network from different types of attacks which are mostly related with the traffic, we have come up an idea to integrate SDN with IoT for better performance, security and access control mechanism.

Here the proposed model is designed into layer wise as SDN separates a network's control plane from its data plane, logically centralizing intelligence to enable high-level network programmability. Our main component of conforming security is "FIS" which is implemented in the IoT Controller as it takes required decision based on data provided by the nodes.

In **Data Plane** the nodes are mainly the devices or sensors which mainly work with the data of different connected devices. These data are passed in IoT Controller where FIS (Fuzzy Interference System object from data) will perform the necessary security measurements. It Mainly works on the basis of **Fuzzification** logic which will analysis the traffic of data flow according to the predefined membership function values and rules of different types of features. Its output will help us to block the malicious packets from entering into the network. It will help IoT Controller to make more efficient decision about forwarding the packets of data. Physical layer of IoT

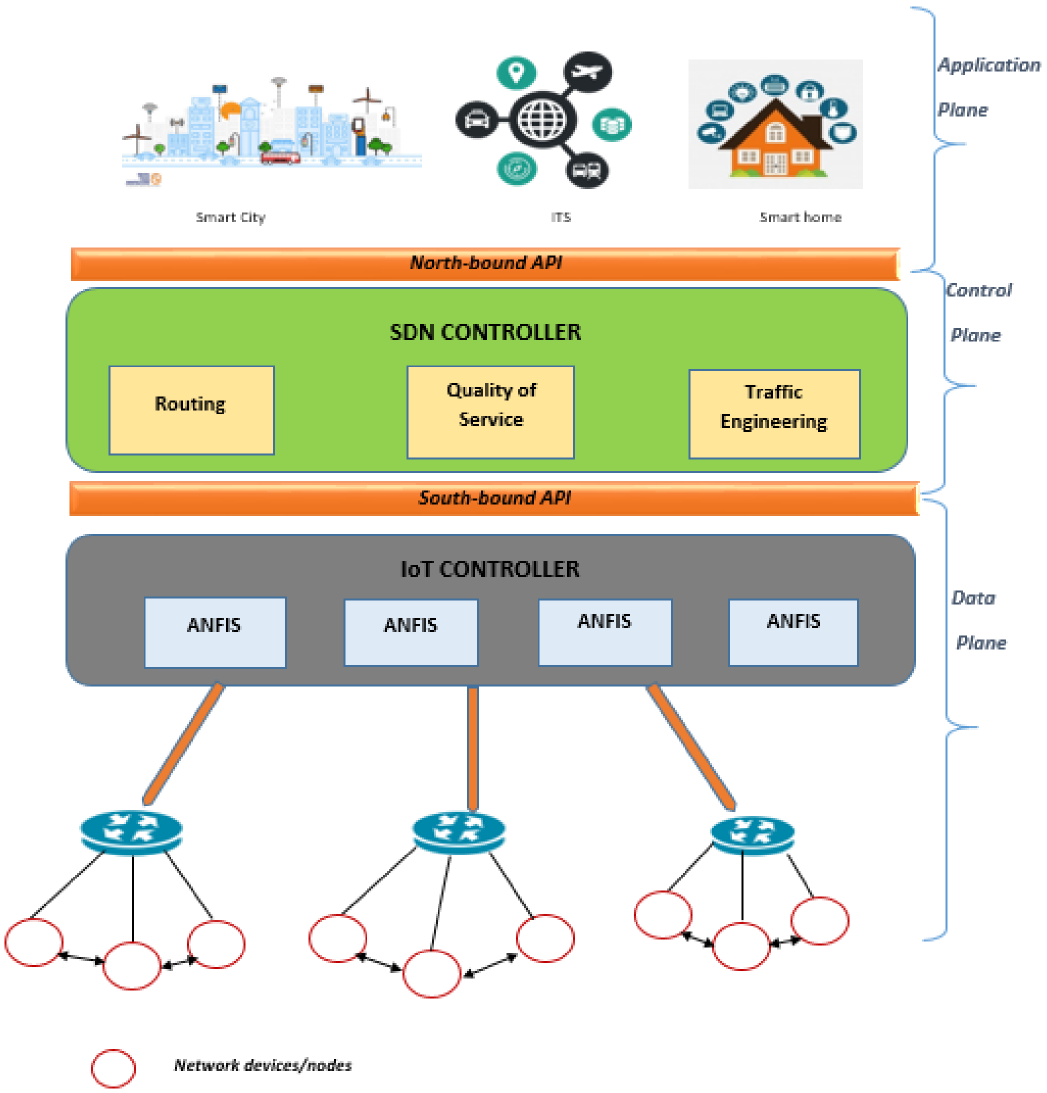


Figure 3.1: System Model

network model and the Infrastructure layer of SDN is embedded together as Data Plane as the functionality is same for both the layers.

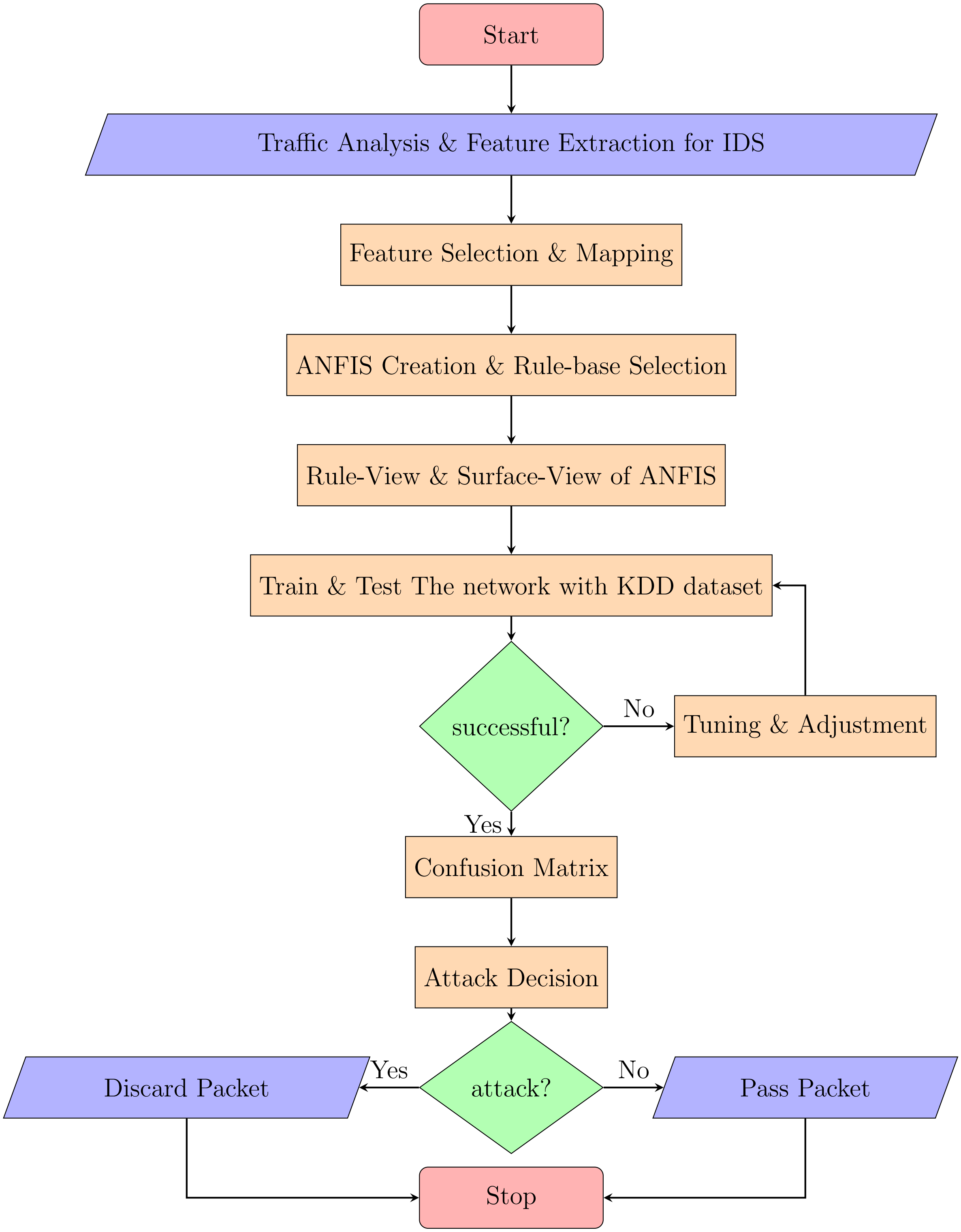
In **Control Plane** the SDN Controller monitors the incoming and outgoing traffic thus it maintains the quality of service and routing data in the Application plane. IoT Controller communicates with the SDN Controller with the rules depending on the networking protocol. So attacks in this control plane will be identified by SDN Controller. It actually behaves like a safety guard which authenticates the network devices and accepts only authorized traffic.

In **Application Plane** IoT based application and services are considered. Security of this layer is provided by SDN **OpenFlow** Algorithm which is shown as the

Northbound API in the architecture.

So by Integrating IoT and SDN network and implementing FIS in Controller data transmission will be protected in every plane and thus it can play an important role in security of IoT Network Model.

3.2 Flow Chart


Start
Traffic Analysis & Feature Extraction for IDS
Feature Selection & Mapping
ANFIS Creation & Rule-base Selection
Rule-View & Surface-View of ANFIS
Train & Test The network with KDD dataset
successful?
No
Tuning & Adjustment
Yes
Confusion Matrix
Attack Decision
attack?
Yes
Discard Packet
No
Pass Packet
Stop

CHAPTER IV

Algorithm Analysis

4.1 Traffic Analysis

A SDN based IoT infrastructure basically provides free-flow of data from sensors and wireless devices and the efficiency of the network depends on the management and security of traffic. Network traffics are dynamic and hence its more prone to malicious attacks such as DDOS, MITM, Replay, Side Channel etc.

4.1.1 Traffic Analysis Technique

There are various classification techniques to classify the network traffic, but among these the following three techniques are mostly used-port based, payload/DPI (Deep Packet Inspection) based and ML (Machine Learning)-technique.

In **Port-based technique**, IP addresses are identified and used to classify the corresponding applications which are registered under Assigned Number Authority (AINA). In the other side, **Payload-based** or **Deep Packet Inspection(DPI)** are basically used to classify dynamic port numbers (peer to peer applications) and packets are analyzed for signatures and authentications of network applications of traffic. **ML (Machine Learning)-technique** uses trained classifiers as input for traffic classification based on the data set.

For our proposed system, to analysis the traffic efficiently we are applying ML-technique, as port-based classification doesn't provide the identification of dynamic ports and payload-based doesn't work for encrypted traffic and requires continuous updating of signature patterns of new applications. ML-technique overcomes these shortcomings of the following classification techniques and works more efficiently to classify data packets.

ML (Machine Learning)- Technique

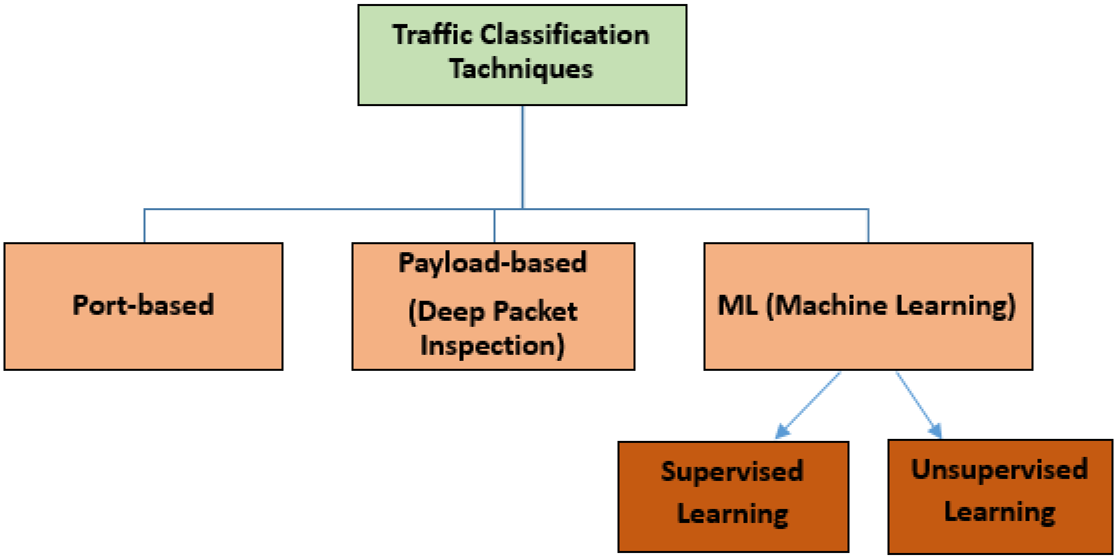

Traffic Classification Techniques
Port-based
Payload-based (Deep Packet Inspection)
ML (Machine Learning)
Supervised Learning
Unsupervised Learning

Figure 4.1: Traffic Analysis Technique

Machine learning is used for dynamic analysis for traffic and uses **WEKA** tool for detection method. It has two techniques for classification- supervised and unsupervised. In **Supervised** technique there is a training data set as input to train the system model for the expected output but in **unsupervised** technique there is no training/known data set and it works based on the prior knowledge or the statistical information.

Supervised technique has so many classifiers in order to classify data such as -Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Logistic Regression (LR), Random Forest, Naïve Bayes, Bayesian Network etc. To achieve our system purpose efficiently we are using the supervised machine learning technique and trained our proposed FIS (Fuzzy Interference System) with the NSL-KDD dataset and the selected classifier [**?**].

4.2 Feature Extraction

By analyzing the network traffic, we get a data set which is the combination of the malware and benign data packets and this is the first major component for any malware detection system. A feature extractor is used to extract the features from the specified data set and we need to extract a group of features to detect attacks, which is not possible by extracting any specific or single feature.

4.2.1 Feature Extraction Tool

Here we are using the **Wireshark** and **Net Mate tool** for the corresponding live data packet capturing and feature extraction purpose.

1. **Wireshark:** Wireshark is an open source software and an efficient network packet analyzer. Wireshark captures the network traffic from various wireless devices and displays them with very detailed protocol information and save the captured data packet. It can also export some or all packets in a number of capture file formats and filter them on many criteria. The basic features of Wireshark tools are-

    - Capture traffics from live network or read data from already captured file.
    - Terminal version, named Tshark or GUI is used to browse captured traffic.
    - Display filter is used to refine and edit traffic programmatically.
    - For dissecting protocols, Plug-ins is developed.
    - Captured traffic can be used to detect VoIP calls when compatible encoding is used for encoding.
    - Only selected traffic appears with several timers, settings and filters.

2. **Net Mate Tool:** After capturing data packets, the features are extracted using Net Mate tool as features depict the behavioral description of traffic. Net Mate includes two types of modules:

    - Packet Processing Modules designed to implement different metrics
    - Export Module that implement different output module

    Our concerned flow features are implemented in **Packet Processing Module**. Two different types of rules are used to produce the output: description rules and recognition rules.

4.3 Feature Selection

Feature Selection is an important step after traffic analysis to detect the abnormality occurring in a system. It can be defined as automatic selection of attributes in data samples that are most relevant to the predictive modeling problem. It does the

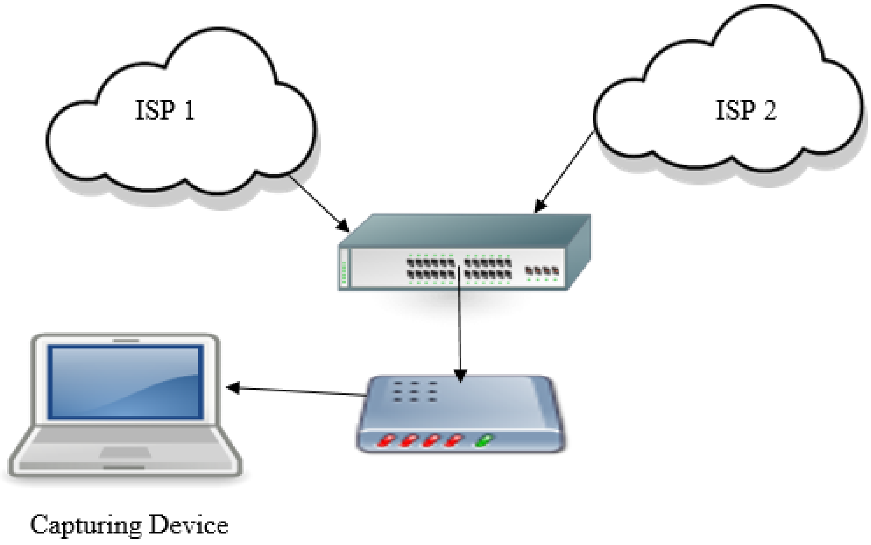


Figure 4.2: Feature Extraction Process

mapping to excludes the irrelevant or redundant attributes and specifically defines most prominent for the better performance of the system.

As in our proposed system we are working on a large amount of a traffic of a network so feature selection should be done for the following requirements:

- To create an accurate predictive model that will give a better accuracy whilst requiring less data.
- it reduces the complexity of a model and makes it easier to interpret the required result.
- It enables the model to train faster on data sample as there is no redundant attributes.
- This method also reduces the problem of over fitting by enhancing the generalization in the model.

4.3.1 Selection Method

There are mainly three methods that are used in feature selection:

1. **Filter Method:** Here features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable which

is Machine Language Independent.**Pearson's Correlation, LDA, ANOVA, Chi- Square** are the methods which are used to define correlation among the features.

2. **Wrapper Method:** This method considers the selection of a set of features as a search problem or algorithm to validate the prediction where different combinations are prepared, evaluated and compared to other combinations. After evaluating it assigns a score based on model accuracy. It can always provide the best subset of features. But this method has a high computational cost.

3. **Embedded Method:** This method tries to combine the efficiency of other two methods and performs the selection of variables in the process of training and is usually specific to given learning machines. It basically learns which features best contribute to the accuracy of the model while the model is being created. Most common algorithms are the **LASSO, Elastic Net, Ridge Regression**used in this method.

For our model **Wrapper Method** is most applicable. As at first we have analyzed network traffic and after that we have implemented a Search process for extracting Unusual features to detect our attacks. From the complete list of Feature set we have further selected the most effective features to make our feature domain more powerful.

4.3.2 Selection Tool

The immediate step after feature extraction of any attack detection procedure is feature selection which is the final input feature set to feed into the system by using any machine learning technique. To select the desired features from the extracted features set, an efficient tool-set, WEKA is used in our attack detection process.

WEKA: WEKA, (Waikato Environment for Knowledge Analysis), named after a flightless New Zealand bird, supports many feature selection techniques, i.e. correlation based, information gain based, learner based etc. Weka is a set of machine learning algorithms for data mining tasks. The algorithms can be used directly on dataset or it can be called from Java code.

Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It provides SQL access with assistance of Java Database Connectivity. Weka provides four UI:

- Explorer
- Experimenter
- KnowledgeFlow
- Simple CLI.

Explorer is the main user interface of Weka which have following panels:

1. **Preprocess:** Choosing the data file.
2. **Classify:** Applying and experimenting with different algorithms on preprocessed data files.
3. **Cluster:** Applying different clustering tools, which identify clusters within the data file.
4. **Association:** Applying association rules, which identify the association within the data.
5. **Select attributes:** Seeing the changes on the inclusion and exclusion of attributes from the experiment.
6. **Visualize:** Seeing the possible visualization produced on the data set in a 2D format, in scatter plot and bar graph output.

The user cannot move between the different tabs until the initial preprocessing of the data set has been completed. This procedure can also be done with component based KnowledgeFlow and from Simple CLI. Experimenter provides option to compare predictive performance of machine learning algorithms on data-sets.

4.4 Feature Specification on Proposed Model

From extraction procedure eighteen features are collected which are grouped into nine features for the convenience of our work.And these nine features are taken as input for the further process.The mapping or selection of features are simplified as the below figure 4.3

1. **Server Overload:** It is one of the most common feature that happens in any kind of physical layer attack in the network. Increase of volume of data packets, excessive traffic is the indication of Server Overload. Though it's a common

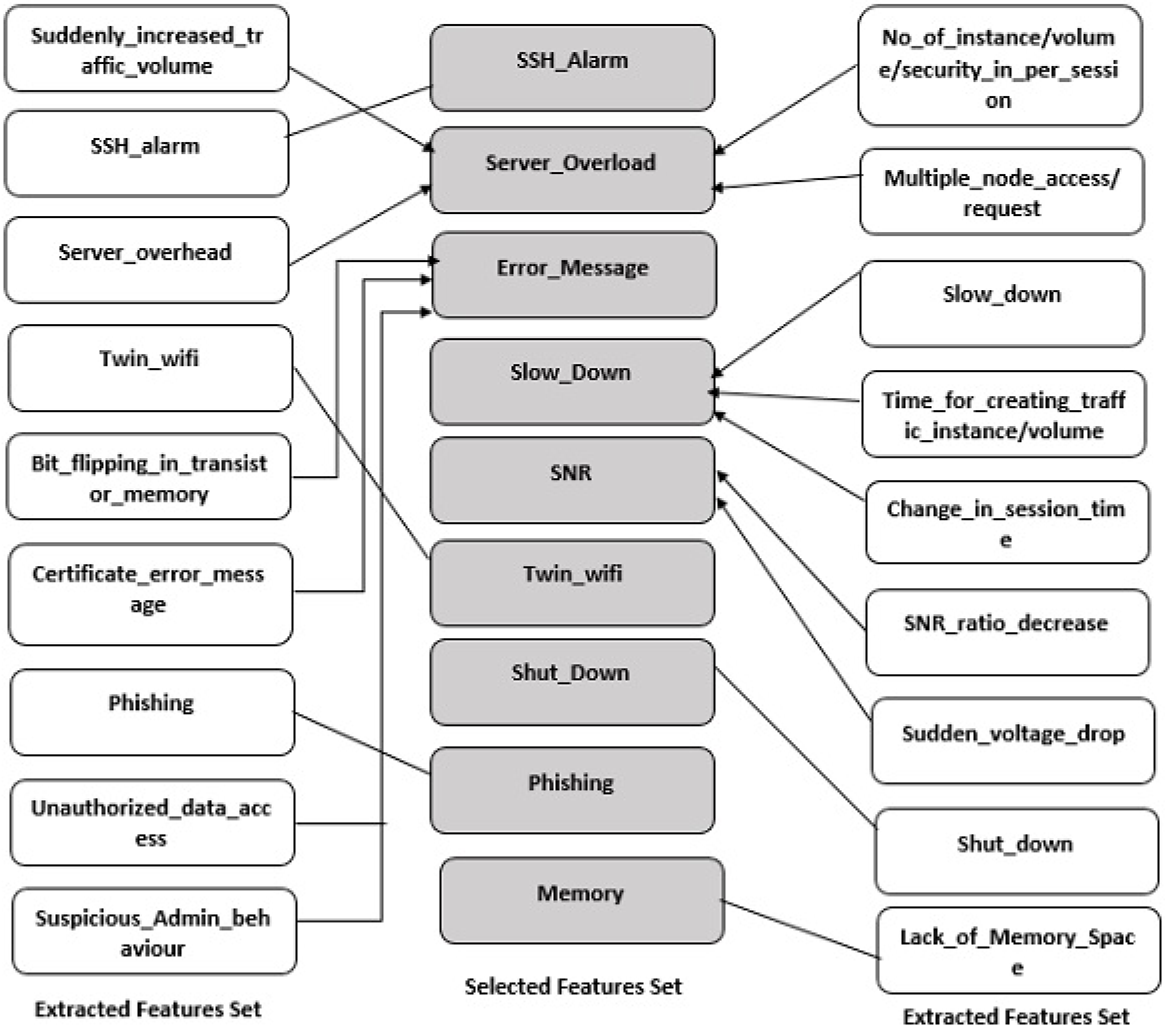


Figure 4.3: Feature Extraction and Selection

feature but there is possibility of Side Channel Attack, Malicious Code and DDoS attack.

2. **Slow Down:** It's a common feature of DDoS and Malicious Code Attack. In DDoS there creates traffic floods in bandwidth and resources so the performance evaluation becomes very poor. It is definitely a symptom that something is wrong with the system. In malicious code there occurs illegitimate actions which creates load on the system.

3. **Sudden shut Down:** It's the extreme case of DDoS attack in the network. a denial-of-service attack (DDoS attack) is a cyber-attack in which the perpetrator

seeks to make a machine or network resource unavailable to its legitimate users by temporarily or indefinitely disrupting services of a host connected to the internet. So when network will not able to manage the overload it will just shut down.

4. **Error Message:** It is the most common characteristics of attacks that frequently occurs in IoT network model. It will generate automatically from the Operating System when it will suspect unusual activities in the network. So it is a great source of predicting that there is a third party in the network who is trying to do something illegal in the network. Features-Bit flipping in memory cells, Suspicious Admin Behavior, Unauthorized Data Access are redundant which is mapped to exclude after feature selection method. Because all these three activities are unusual and it will result an error message. DDoS, Side Channel Attack, Malicious Code Attack, Man in The Middle(MITM) attack can be suspected by this feature.

5. **SSH Alarm:** SSH is one of the most popular communication protocols on the Internet used by admins, developers. SSH alarm is an email alert, when someone logs server via SSH (Server Secure Shell) can be pretty useful to track who is actually using server. It's a very unique feature to track MITM attack as an intruder might not login at first attempt.

6. **Twin-WiFi:** In MITM the main aim of an intruder is to entry the network and hampers the integrity, confidentiality, authenticity of admin. To get illegal access he can adapts the method of duplicate WiFi SSID or Address that is a very prominent feature to identify that the system is being attacked by the third party.

7. **Phishing:** It's a great threat to the security of users. It is actually a cybercrime in which targets are contacted by email, telephone or text message by someone posing as a legitimate institution to lure individuals into providing sensitive data such as personally identifiable information, banking information, credit card details, and passwords. Intruder who conducts MITM attack mostly does this to earn in an illegal way.

8. **SNR decrease:** When noise of a system increases the SNR decreases that indicates the poor performance of the system. Voltage drops with proportional to SNR, that is not definitely a good symptom for a model. It is the most

prominent feature to detect the Side Channel Attack. it is caused by the information gained from the network so the noise increases which should be noticed to detect attack.

9. **Lack of Memory Space:** Malicious code is an application security threat that cannot be efficiently controlled by conventional antivirus software easily. It describes a broad category of system security terms that includes attack scripts, viruses, worms, Trojan horses, backdoors and malicious active content. So sometimes it suddenly just occupies the memory space of user device and gives warning to the user of “Memory is Full”. That’s definitely occurs a great problem of storing.

   Here FIS will primarily work on these **Nine** Selected features where rules will be considered in controller to identify DDoS, MITM, Malicious Code Attack and Side Channel Attack. Rules are defined according to the priority of the features.

CHAPTER V

Performance Analysis

5.1 Fuzzification

5.1.1 Fuzzification Method:

Fuzzy logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1, considered to be **"fuzzy"** . But in **Boolean logic**, the truth values of variables may only be the "crisp" values 0 or 1. So Fuzzy Logic resembles the human decision-making methodology, it gives opportunity to take decision more efficiently and accurately. It's very flexible and mathematically easy and also has the functionality to create customized input and output data set which contain all the possible elements of concern in each particular application. by designing necessary mapping function which makes it more adaptive system. There are some basic unit of FIS:

- Fuzzification Unit
- Knowledge Based Rules
- Decision or Controller Unit
- Defuzzification Unit

CHAPTER VI

Future Work & Conclusion

6.1 Future Work

IoT is a large field where we have focused on Security but there are many more to work or analysis in future to make this network most secured and efficient to use in everyday of life. Though we have tried to make an effective model to protect the network from the base level but still there are a lot of scope to work further. Only Perception layer's attack has been considered here but it can't detect all attacks it only can detect – **"DDoS, MITM, Side Channel Attack, Malicious Code Attack"**. So in future more attacks should be analyzed to detect for more security measurements. Proposed model has not been analyzed for time complexity so can't tell about the time efficiency of the system. Here only concept of Firewall has been used but for a perfect security software or hardware program an Adaptive Firewall should be designed in future.

6.2 Conclusion

There are many IDS model for IoT network. But here we have considered a SDN based IoT network model where SDN provides the security of network and Application layer. So here an efficient, simple human perception rule based FIS has been designed to protect the physical layer from passing any kind of malicious network traffic. The FIS has been designed in a centralized manner by using concept of ML to be implemented in SDN Controller. Through this model the four most common attacks of IoT perception layer has been detected which has not been detected in the same way before. Further analysis can be done to make this model more efficient to use in a real network model.

References

[1] V. V. Vijayan and C. Anjali, "Prediction and diagnosis of diabetes mellitus — a machine learning approach," in *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 122–127, 2015.

[2] Jurafsky, Daniel, Martin, and J. H., *Speech and Language Processing (2Nd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009.

[3] "K-nearest neighbours." `https://www.geeksforgeeks.org/k-nearest-neighbours/`.

[4] Evgeniou, Theodoros, Pontil, and Massimiliano, "Support vector machines: Theory and applications," vol. 2049, pp. 249–257, 01 2001.

[5] D. V. Patil and R. S. Bichkar, "Article: Issues in optimization of decision tree learning: A survey," *International Journal of Applied Information Systems*, vol. 3, pp. 13–29, July 2012. Published by Foundation of Computer Science, New York, USA.

[6] R. D. S. Raizada and Y.-S. Lee, "Smoothness without smoothing: Why gaussian naive bayes is not naive for multi-subject searchlight studies," in *PloS one*, 2013.

[7] "How the random forest algorithm works in machine learning." `https://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/`.

[8] "Linear discriminant analysis – bit by bit." `http://www.grroups.com/blog/linear-discriminant-analysis-bit-by-bit/`.