Islamic University of Technology (IUT)

INTRUSION DETECTION IN IoT BASED SYSTEMS

Supervisor

A. B. M. Ashikur Rahman

Asst. Professor,

Dept. of CSE, IUT

A thesis submitted in partial fulfilment of the requirements
for the degree of B. Sc. Engineering in Computer Science and Engineering

Academic Year: 2019-2020

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
A Subsidiary Organ of the Organization of Islamic Cooperation (OIC)
Dhaka, Bangladesh

March 17, 2021

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by under the supervision of A. B. M. Ashikur Rahman, Asst. Professor in the Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Abdoul Bagui Alh Boubakari

____

Student ID - 160041086

Aly Abdelkader Gelany

____

Student ID - 160041087

Abdul Aziz Yousufzai

____

Student ID - 160041088

Hamdan Harir

____

Student ID - 154452

Approved By:

Supervisor:

A. B. M. Ashikur Rahman
Asst. Professor
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), OIC

Acknowledgement

The success of this thesis is the fruit of the various supports and help from many individuals. We would like to express our sincere gratitude to all of them.

We would like to express our grateful appreciation for **A. B. M. Ashikur Rahman**, Asst. Professor, Department of Computer Science & Engineering IUT, for being our advisor and mentor. We also do appreciate his wonderful advice for the documentation of this thesis.

We are grateful to **Prof. Dr. Abu Raihan Mostofa Kamal**, Head of the department, Computer Science and Engineering (CSE) for his enormous support.

We are also grateful to our parents for all their care, motivation and support throughout this long journey.

Abstract

The constant evolution of Technology has led to a huge number of devices connected over the Internet and sharing sensitive data forming the IoT network. By its architecture and configuration, the IoT network is more vulnerable to attacks and has become the main target for attackers who constantly try to create new forms of attacks to overcome existing security measures. To mitigate the risk of intrusion and attacks, there is the necessity of developing intelligent security systems that can efficiently distinguish between normal traffics and attacks and capable of reporting these attacks. For this, we have proposed a hybrid intrusion detection system capable of reporting previously seen and unseen form of attacks. Our model uses a combination of feature transformation and protocol Analyzer to enhance data quality and to avoid unnecessary computation overhead. The enhanced data is then fitted to some machine learning algorithms namely K-Nearest Neighbour (KNN), Logistic regression (LR), and Multilayer Perceptron (MLP).

Keywords: Intrusion detection system (IDS) . Outlier detection (OD) . Protocol Analyzer . Features Tranformation. Hybrid IDS

Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview

In our modern lives, we are surrounded by electronic devices that assist us by reducing the time taken to perform tasks or to achieve what is not humanly possible. Recently, general-purpose electronic devices (Phones, Air conditioners, Printers, etc...) have gained the functionality of being connected to the internet enabling transfer of data within each other. This ideally allows modern devices to be more capable and less reliant on direct human input thus, resulting in a larger scale in what is known as IoT. IoT is an umbrella of technologies to connect diverse devices (wireless sensors, Phones, fridges) to the Internet. They graciously provide users with digital services in various application domains such as remotely monitoring patient's health and controlling appliances in smart homes. Having such perks and being applicable in many domains resulted in a surge in the number of IoT capable devices. They by design gather data from the surrounding environment which some might be sensitive and private then, share it and interact with each other which introduces multiple points of attack. IoT devices offer very limited security features which make them vulnerable to new attacks. Therefore, we see the imperative necessity of having an Intrusion Detection System. An Intrusion Detection System or IDS for short, Is a Layer of security that enables the prevention of Cyberattacks by detecting, classifying and reporting them.

1.2 Problem statement

,

One of the main weaknesses of signature-based techniques is the inability to accurately detect previously unseen attacks on which the model was not trained while the anomaly-based techniques suffer from high false-positive. As we know that attackers constantly create new forms of attacks to overcome existing security measures, the proposed method should be an effective intrusion detection model that can accurately detect previously seen and unseen attacks with a very low false-positive rate.

1.3 Motivation and scope of research

Firstly, the Exponential growth of IoT devices in critical infrastructures such as Healthcare, Autonomous vehicles, Nuclear power plant has made the IOT network the main target for hackers. A compromised access to these infrastructures can lead to serious environment damages or privacy violation and information thief. Many researches have been done using signature and anomaly-based techniques.
Secondly, An Hybrid based techniques can combine the benefits offered by both approaches and give higher accuracy that can more effectively detect intrusion in IOT based Systems.
Thirdly, Developing an intrusion detection system that can run on edge devices. The cloud-centric approach of IoT has led to many challenges such as Bandwidth limitation, data storage, data processing, and security issues. Since all the data produced by the rising number of sensing devices need to be sent over the network to the cloud for processing. For the security concern, given that the connection type used in IoT is mainly wireless, the possibility of intrusion and attacks on the network is getting higher. Hence, there is a crucial necessity of moving from the cloud computing paradigm to the edge computing paradigm to mitigate the above-mentioned challenges.

1.4 Research Challenges

Although a lot of progress has been done in the field of Security in IoT, many key points still remain challenging. Categorizing attacks is a challenge for any intrusion detection system as new types of attacks are being developed constantly to overcome existing security measures. Additionally, the availability of real dataset is a limiting factor for research in the field of IOT network security, as many of the publicly available datasets are synthesized in the lab using specific software. The available datasets also suffer from poor quality and require a lot of preprocessing.

1.5 Thesis Outline

The rest of this work is organized as Follow. In chapter 1 we have discussed the scope and challenges of our study. While In chapter two, we presented the literature review and the background study on the Intrusion detection system. In chapter three we discussed our proposed model in details. Chapter 4 explained the performance analysis of the proposed model and Finally, we finished with the conclusion and future work in chapter 5.

2 Background study

Intrusion Detection Systems (IDSs) are considered well known tools for monitoring and detection of malicious traffic in communication networks. However, IDS is a technology that uses highly developed and complex algorithms for processing large volumes of data [11].The complexity of the algorithms results in long computation time. IDS captures network traffic in real time and compares the received packet patterns with known patterns to detect anomalies in network. Yet the cost and high processing time to handle traffic load is a challenge in IDS.

To overcome the issue a lots of work has been done in the part, various techniques have been proposed to improve intrusion detection in IoT based networks, mainly there are three methods for Intrusion detection system (IDS), as we can see in figure 1 [1]. Each of these three techniques has some pros and cons, in this part we will go in deep in each of these methods and we will summarized some of those papers which used this methods.

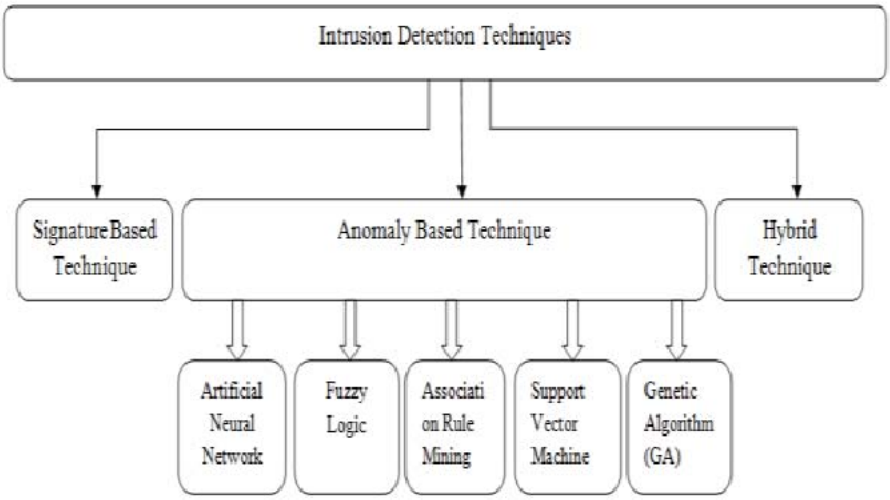


Figure 1: Intrusion Detection Techniques.[1]

2.1 Signature based techniques

Signature based intrusion detection is performed by comparing set of known attacks with newly inputs if it matches the signature or known attack it will be alerted as an intrusion otherwise normal data. set of known attacks or signature are saved in a database.as we can see in figure 2 [2]

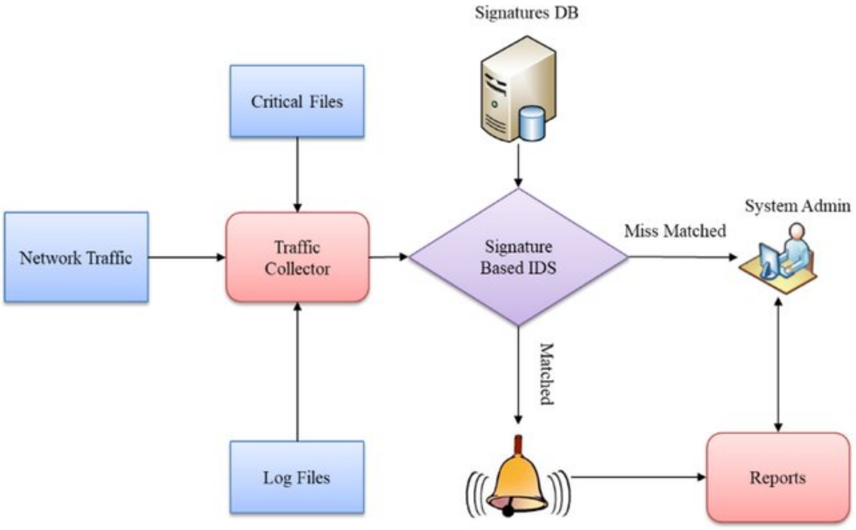


Figure 2: Signature Based Intrusion Detection Technique.[2]

Robert P. et al [12] proposed an intrusion detection system called Clust-IT, based on clustering algorithm. They proposed a novel approach established on density-based clustering, for effective intrusion detection using OPTICS (ordering points to identify the clustering structure) algorithm, taking into account the heterogeneity and resources constrain of the IoT network. The proposed system runs on the edge of the IoT network such as a router, where a series of cluster algorithms are applied to the data collected by the different smart devices at the endpoint of the network. This approach stands out from other machine learning techniques in that it does not require labeling of data, or prior knowledge of the type of anomalies, as it is the case with signature-based detection techniques.

The proposed method has advantages such as interpret ability since it produces a model easy to understand and manipulate. Also, as an advantage, we can cite the adaptability, since the proposed model is build based on normal behavior in

the system, making it possible to detect any form of anomaly or outlier without previous knowledge of it. Albeit the fact that CLUST-IT has shown significant results, compared to other state-of-the-art algorithms such as simple k-means, it suffers from some drawbacks such as overhead for generating clusters. As the runtime for generating the cluster may be high compared to other simple clustering algorithms. Another major drawback is the inability to assess the cluster quality formed by the system.
Zheng et al[13] Introduce IELCA algorithm for intrusion detection in IoT Devices, they first of fall normalize the collected data before reduction and classification by using Z-score normalization technique. They used similarity measure function for high dimensional data as the weight to improve the between class scatter matrix. Then they combine Linear Discriminant Analysis (LDA) with IELCA to maximize between-class and minimize within the class distance, for obtaining the optimal transformation matrix and reduced dimensions original data. At end they aggregate IELCA with Extreme Learning Algorithm (ELM) classification algorithm for classification and determining the security level of IoT device. The advantage of proposed algorithm is high accuracy 92.35 and detection rate of 91.35. Disadvantage is, its doing all the process and computations in base station rather than edge node.

Eskandari et al.[14] presented Passban as an intelligent IDS, directly hosted and executed by an Edge device, a Two different Scenarios configured Simulation of a home automation system is built to asses the performance of IDS both threat detection resources usage wise, In Scenario 1 the IDS is directly deployed and executed on the IoT gateway: in this case, Passban is able to protect the latter and all the IoT devices directly connected to it. In Scenario 2, Passban is provided as a separate add-on device independently connected to the network it has to protect: in this case, it can monitor incoming and outgoing network traffic of the IoT devices connected to the gateway, inspecting them for suspicious patterns occurring in the net- work traffic. they implemented passband using two classification techniques iForest and LOF evaluated it against common attacks namely port Scanning, HTTP brute force, SSH brute force SYN flood attack.

their experiments reveled that using iForest, Passban IDS can achieve F1 scores greater than 0.9 on some attacks (0.99 best case and 0.79 worst case). In terms of resource utilization, we proved that Passban can be executed even on cheap IoT gateway boards.

Alaiz-Moreton et al[15] proposed ensemble methods and deep learning models, for multiclass classification in IoT network that uses the MQTT protocol. They used these supervised learning techniques to classify a series of anomalies from normal data obtained in a test environment with several sensors, actuators, and a server. The anomalies were mainly denial of service (DoS), Man in the middle (MitM), and Intrusion attacks.in the proposed work, three datasets were used corresponding to each form of attack, on which after combination and data preprocessing, they trained different model: XGBoost (extreme gradient boosting) [16] fast-histogram algorithm for ensemble methods, LSTM and Recurrent GRU (Gated recurrent units) [17] for deep learning. The results of the experiment showed that ensemble method (XGBoost) perform well more than its concurrent linear m sethods GRU and LSTM. Notwithstanding the advantage of high accuracy observed, this approach has a major drawback which is non-adaptability. Since the models are only trained on known anomalies. Furthermore, this method does not solve the bottleneck problem on IoT networks related to the cloud computing paradigm.

Zaffar et al[18] proposed a system based on the edge computing paradigm using a gateway device located on the IoT edge. This system was capable of detecting a rare event from the data sensed by the end nodes using unsupervised learning algorithm, thus escaping the necessity of labeling the incoming data.

This paperwork contributed by proposing an edge computing framework capable of performing analytic computation at the gateway level and detects rare events from sensed raw data. Thus, reducing the burden on the cloud computing system and the cost of data transportation. Another contribution is the usage of unsupervised learning approach, micro clustering and macro clustering for the continuous data stream processing without the need of using storage resources, and also making the detection of rare events possible without prior knowledge of them.

the proposed framework can be divided into stages. At the first stage We have sensing devices that continuously sense sound in an environment and produce continuous waveform. The sensed data is sent to a second stage where they are buffered in the local memory of the gateway device. At stage 3, the buffered data is supplied to a data framing module where it is further decomposed into smaller frames ready for feature extraction. Then comes one of the most important stages that is feature extraction. At this stage, the time domain and frequency domain of the previously framed data are extracted. The main features are the Linear predictive coding (LPC) to preserve the time domain. They also extracted the Mel-frequency cepstral coefficients (MFCCs) and Gammatone frequency cepstral coefficients (GFCC). the combination of these features uniquely identifies the incoming signal to identify rare-event. At the last stage, the extracted features are then fed into a two-stage unsupervised machine learning algorithm. The first stage of the unsupervised approach is micro clustering using the BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm that produces several microclusters. In the later stage that is macro clustering, the produced micro clusters are merged based on the centroid distance to produce larger clusters. This merging process is done recursively until only two clusters remain, the normal event cluster and the rare-event cluster such as Gunshot, Glass break, Siren, and Scream.

Among the advantages of the proposed system, we can mention the rare even detection without the need for labeling data or having prior knowledge of the rare event.This because the system uses an unsupervised learning approach. Furthermore, we can also talk about the deployment of the system on an edge device, hence reducing the burden on the cloud server, mitigating the data transportation and delay problem.

On the other hand, the system has some drawbacks among which the overhead in determining the thresh hold value for different rare-event. For each rare event, there is a necessity of evaluating the right thresh hold and the frame size for optimal results. Also the fact that the number of false negatives increases as the windows size increases.

2.2 Anomaly based techniques

Anomaly based methodology works by comparing observed activity against a baseline profile. The baseline profile is the learned normal behaviour of the monitored system and is developed during the learning period were the IDPS learns the environment and develops a normal profile of the monitored system. This environment can be networks, users, systems and so on.[2] as we can see in following figure 3 [3]

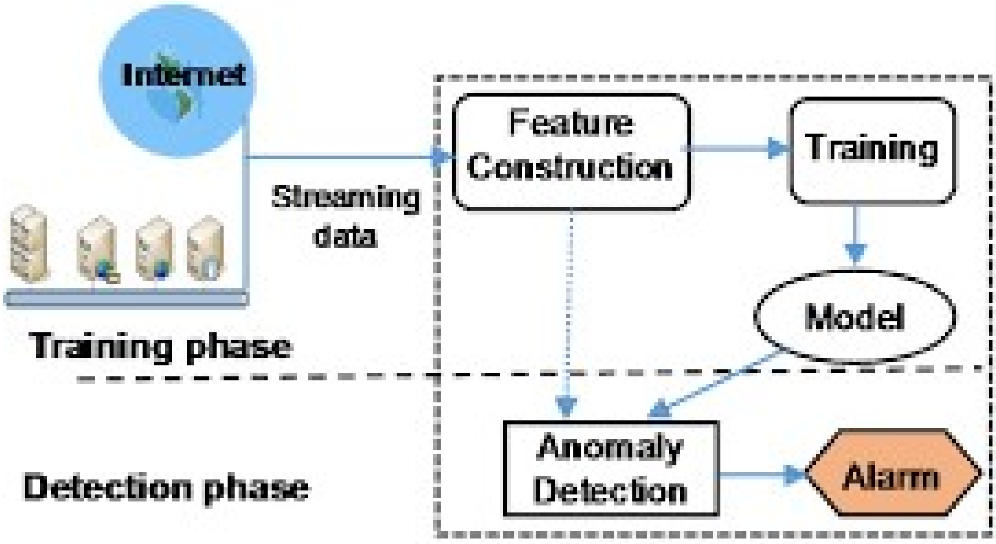


Figure 3: Anomaly Based Intrusion Detection Technique.[3]

With the increase of IoT technology across different industries such as health care, social domain and smart cities, the security of the IoT have become a challenging task especially in the area of intrusion detection. Detecting attacks with low frequency such as U2R (user to root), R2L (remote to local) became a hard task to achieve. In this sense, Hamed, Reza, Raouf, Ali and Kim [19] proposed a Two-layer Dimension Reduction and Two-tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks.The proposed method works as follows. First the NSL-KDD dataset goes through two-dimension reduction process which are principal component analysis and linear discriminative analysis to reduce the high dimensional dataset into a lower one with lesser features. In PCA the eigenvalues and eigenvectors of the data are computed and the eigenvectors with lower eigenvalues are dropped because the contains least information about the distribution of the data. After completing the PCA process, it then continues with the LDA to reduces the features in order to apply the labelled data in an optimal transformation to new dimension. After this process, we can now apply our classification method. The Naïve bayes classifier is applied to the reduce dataset to classified the anomalous behavior which is then refined to normal instance using the certainty-factor version of the KNN. The benefits of this method are that it performs better than the previous existing method in case of detecting low frequency attacks such as U2R and R2L attacks and it also performs better for predicting unknow attacks. Secondly the K-d tree is used to help KNN search faster than traditional approach. Lastly the complexity of the proposed method is low since the data has been reduced using PCA and LDA. However, there are some disadvantages of the proposed method. The proposed method performs low compared to others existing method in terms of detecting high frequency attacks such as Probe and DoS. And also, this method can only be applied in the network layer.

Mudgerikar et al.[20] proposed anomaly-based system level intrusion detection method for IoT devices.E-Spion profile IOT device based on their behavior by getting advantages of system level information e.g. running processes parameters and their system calls.Which is portable with several IOT devices architecture and Linux distribution. Due to minimize the computational/storage overhead they employ edge split architecture. In order to increase the flexibility and reliability it developed 3 layers(PWM,PBM,SBM) of detection with various level of trade-offs depend on detection efficiency and overhead cost.E-Spion Splits architecture into device edges, the bulk of computations done in server side and minimal work is used to be done in IOT devices.

Niedermaier et al.[21] proposed distributed IDS for industrial and sensor applications, based on statistical approach. which is working better on low performance Micro controller Units (MCU) than signature and rule-based approaches. it has some advantages like, dynamic learning without fixed rules, no need for periodic signature in malicious software.they moved IDS into edge nodes of IoT devices. they consider two types of attackers local and remote.To have trusted comparison base, it has to be guaranteed that there is no attacker in network during learning phase.

2.3 Hybrid based technics

Hybrid IDS has been proposed to overcome the shortcomings of SIDS and AIDS, as it brings together two or more of the other methodologies identify both unknown and known attacks. Novel techniques were used to combine the results of SIDS and AIDS.usually SIDs store history of previous known attack and AIDs used to profile normal nodes. as it shown in figure4 [4]
M.Monshizadeh et al.[11] have proposed Hybrid Anomaly Detection Model (HADM) which is a platform that uses a combination of linear and learning algorithms combined with protocol analyzer. The linear algorithms filter and extract distinctive attributes and features of the cyber-attacks while the learning algorithms use these attributes and features to identify new types of cyber-attacks. The protocol analyzer in this platform classifies and filters vulnerable protocols to avoid unnecessary computation load. The linear algorithm initially defines whether the packets are safe or unsafe regardless of the suspected attack type, then extracts the features of the suspected attack and provides them to the learning algorithm. The learn-

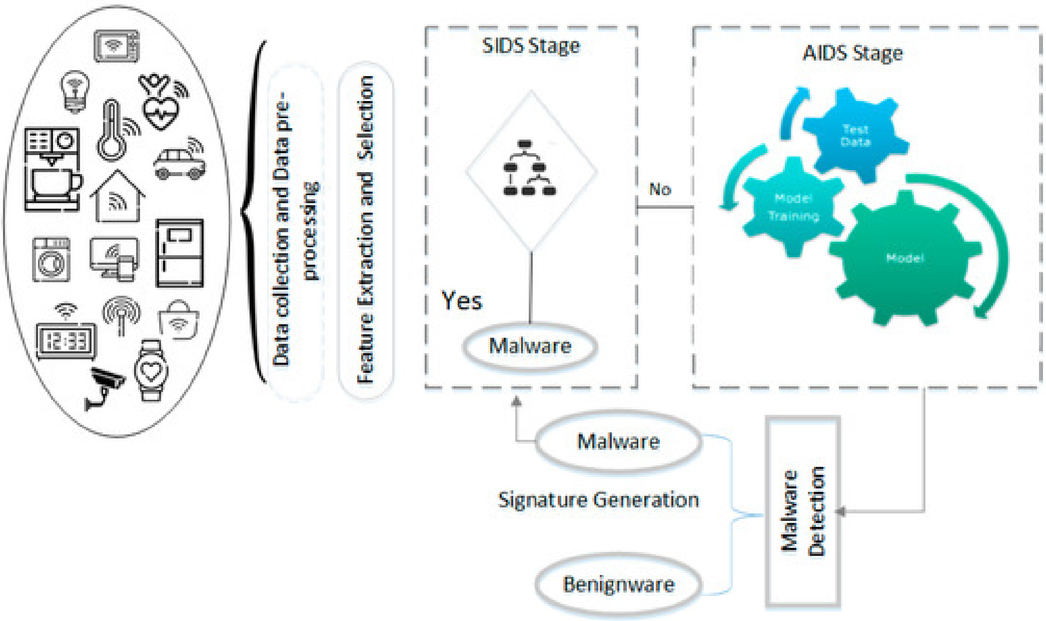


Figure 4: Hybrid Intrusion Detection Technique.[4]

ing algorithm compares the extracted features against known attack features and classifies the suspected attack as either known or unknown. In case of unknown attack, they are labeled. The information about attack is then shared to the validator and database component. The validator and database component validate the output of the linear and learning algorithms. If the actual output differs from the expected output, then the actual output is considered as an error. The have used multiple algorithms for learning and linear with five different datasets. before feeding the dataset into the linear and learning , the have used a bunch of different features selection algorithms in order to reduce the features of the datasets.

For improving system security. Alghayadh et al.[22] Used two tier Hybrid Intrusion detection System (HID), in first tier they try to examine all the network request which are coming from user side by using machine learning techniques e.g. Random Forest, Xgboost, decision tree and KNN, analyzed the Model with CSE-CIC-IDS2018, and NSL-KDD datasets, which given accurate result, in second tier they try to extract various pattern from different users based on user behaviors through numbers of sensors by using Misuse algorithm ,they employed on CASAS datasets which well classify legitimate request. There are limited number of legitimated request for each user, there are no sensors for individual users which may cause error, each sensor is expected to receive request for certain user and fixed duration of time other request will be implies anomaly.

Dr. S. Smys and all Propose Hybrid IDS which focuses on multiple attack detection through use of convolutional neural network model which is a deep learning model for data classification that performs better with diverse database collections combined with long short term memory (LSTM).

The model is divided into four phases collection of data, pre-processing data, training the network and attack identification. Data is input to training model by defining the convolutional layer, size of sliding window, neuron link weights and outputs. Finally in the detection phase, the trained data and actual collected information are processed together to obtain the weights and the training period is used to detect the attacks.z With the use of UNSW NB15 data set combined with validation ratio of 70% for training and 30% of test validation, the proposed system system tested with tensor flow extracts features from dataset and identifies both attacks and normal conditions. Compared to RNN based IDS model the proposed model (HCNN) has better detection performance and ration of true and false positive combined with better efficiency of 98% which is 3% greater than conventional recurrent neural network model (RNN), which makes it suitable for different IOT environments.

3 Proposed Approach

In this section , we are going to discuss our proposed method which is a hybrid model that used the combination of feature transformation, feature selection, protocol analyzer and some machines learning algorithms.

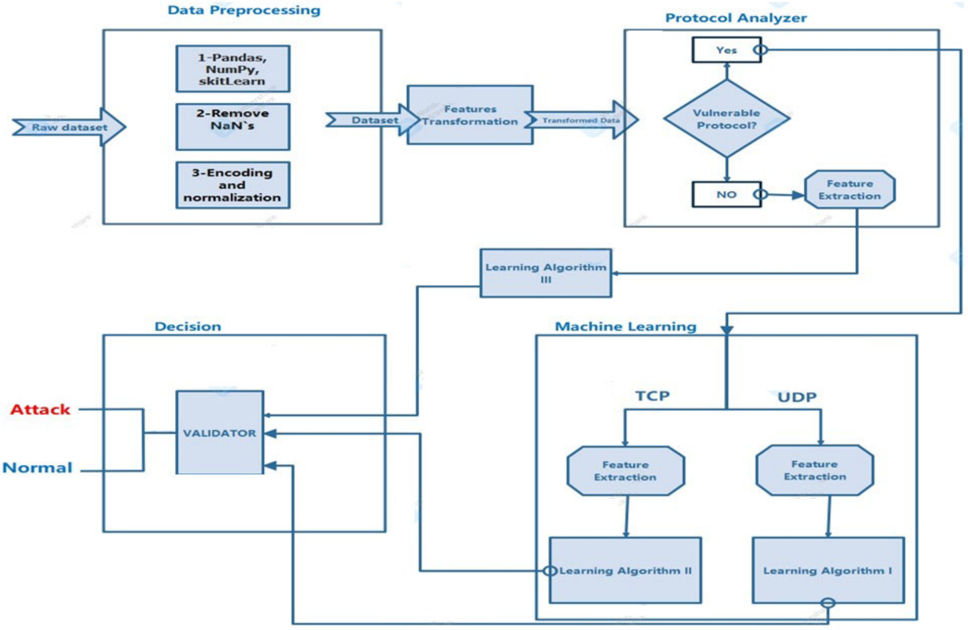


Figure 5: The Proposed Approach

3.1 Data Preprocessing

Data preprocessing is an important step in training a machine learning model. The phrase ”garbage in, garbage out” is particularly applicable to data mining and machine learning projects. Data-gathering methods are often loosely controlled, resulting in out-of-range values (e.g., Income: 100), impossible data combinations (e.g., Sex: Male, Pregnant: Yes), missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis. Often, data preprocessing is the most important phase of a machine learning project, especially in computational biology. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Data preprocessing includes cleaning, Instance selection, normalization, transformation, feature extraction and selection, etc. The product of data preprocessing is the final training set.

The algorithms need data normalization where numeric attributes are transformed into nominal attributes to improve the performance of the algorithms. The IP address and hexadecimal Medium Access Control (MAC) address of the applied datasets are transformed into separate numeric attributes. Each numeric attribute is normalized between 0 and 1 by calculating batch mean and standard deviation, unless there is an already defined range (e.g., IP address range).

3.2 Feature Transformation

Data transformation is the process of converting data from one format or structure into another format or structure. It is a fundamental aspect of most data integration and data management tasks such as data wrangling, data warehousing, data integration and application integration. Data transformation can be simple or complex based on the required changes to the data between the source (initial) data and the target (final) data. Data transformation is typically performed via a mixture of manual and automated steps. Tools and technologies used for data transformation can vary widely based on the format, structure, complexity, and volume of the data being transformed.

As data quality is an essential determinant to intrusion detection, to build the intrusion detection model, we should first conduct feature transformation on the original data to obtain high quality data. we know that for a feature extracted from network data, if normal and the malicious activity share similar characteristics, it is difficult to make a distinction and understand the decision operations.after the feature transformation, the difference between the normal and the attacks will be magnified which makes it easy to separate .the newly transformed data is much more concise and informative. Figure 6 shows the data distribution before and after the transformation.

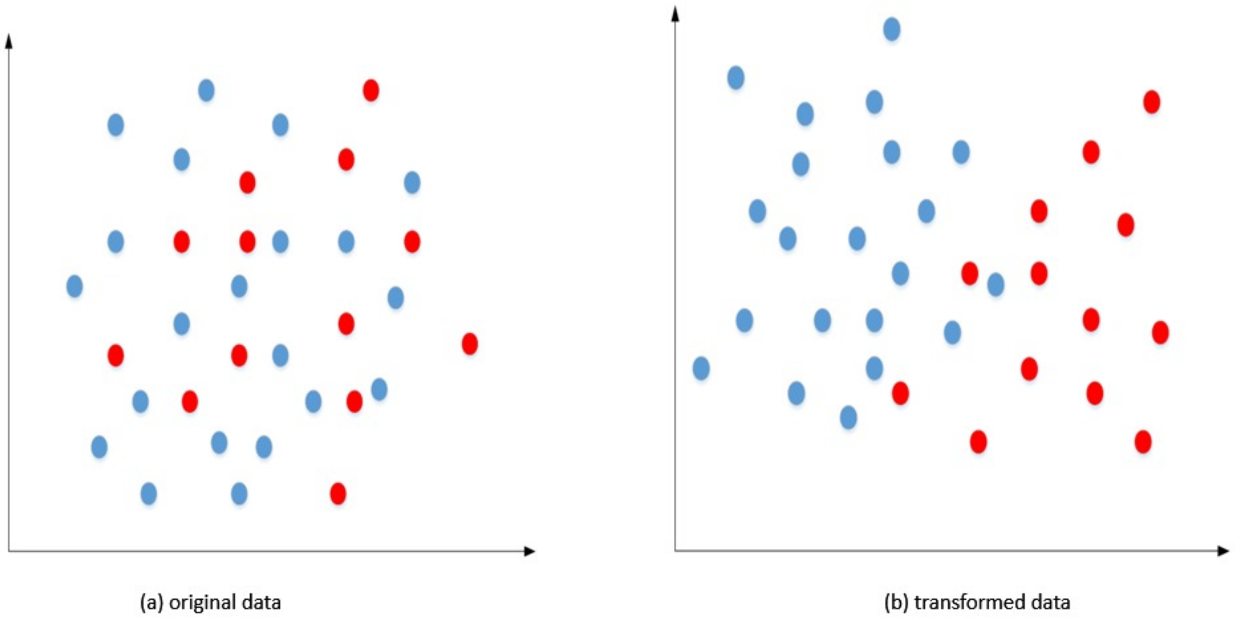


Figure 6: Original data (the blue and red circles corresponding to the normal and the malicious activities, respectively)

The procedure of the proposed data quality improvement technique [23] is shown in Figure 7

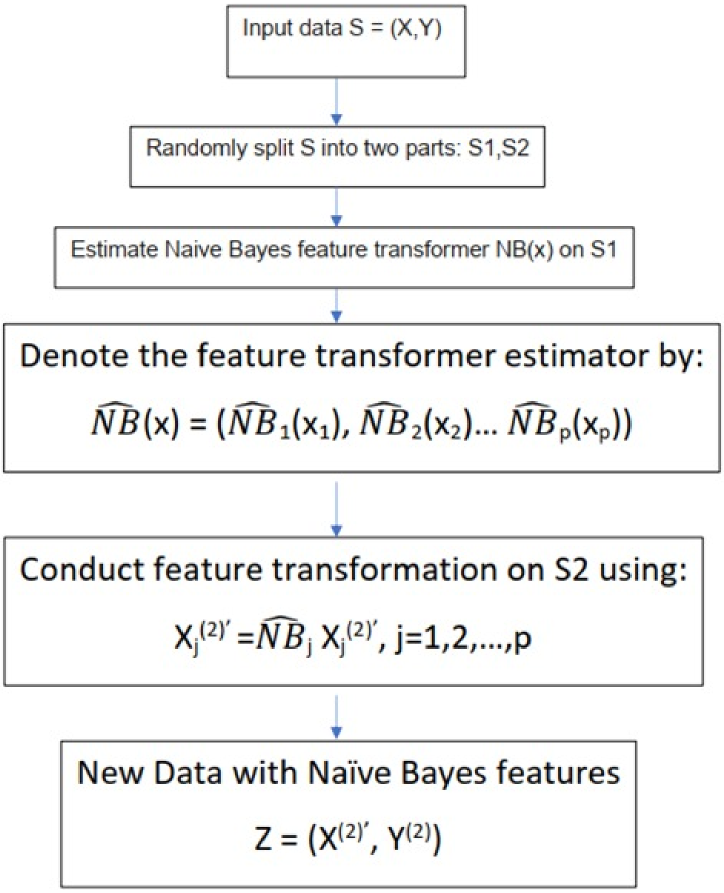

Input data S = (X,Y)
Randomly split S into two parts: S1,S2
Estimate Naive Bayes feature transformer NB(x) on S1
Denote the feature transformer estimator by:
$\widehat{NB}(x) = (\widehat{NB}_1(x_1), \widehat{NB}_2(x_2)\ldots \widehat{NB}_p(x_p))$
Conduct feature transformation on S2 using:
$X_j^{(2)'} = \widehat{NB}_j\, X_j^{(2)'}$, j=1,2,...,p
New Data with Naïve Bayes features
$Z = (X^{(2)'}, Y^{(2)})$

Figure 7: Naive Bayes Feature transformation Equations

3.3 Protocol Analyzer

The mentioned components are deployed in conjunction with one another to filter packets on the communication networks, such as mobile networks and for certain network protocols that are known or considered to be vulnerable to or used in cyber-attacks [11]. This allows our method to expend a smaller amount of processing resource on other network protocols, such as streaming protocols that are not normally vulnerable and thus not typically targeted by cyber-attackers. The ability of our method to focus on vulnerable network protocols helps to avoid burdening network servers with unnecessary computational load. The protocol analyzer filters the network packets and identifies vulnerable protocols. as we know that some protocols such as UDP and TCP are vulnerable, in our method we have classified them into three categories which are TCP,UDP and the others protocols . if the protocol is either TCP or UDP , we sent it for further processing otherwise we just sent it to the selection feature to reduce the computational load.

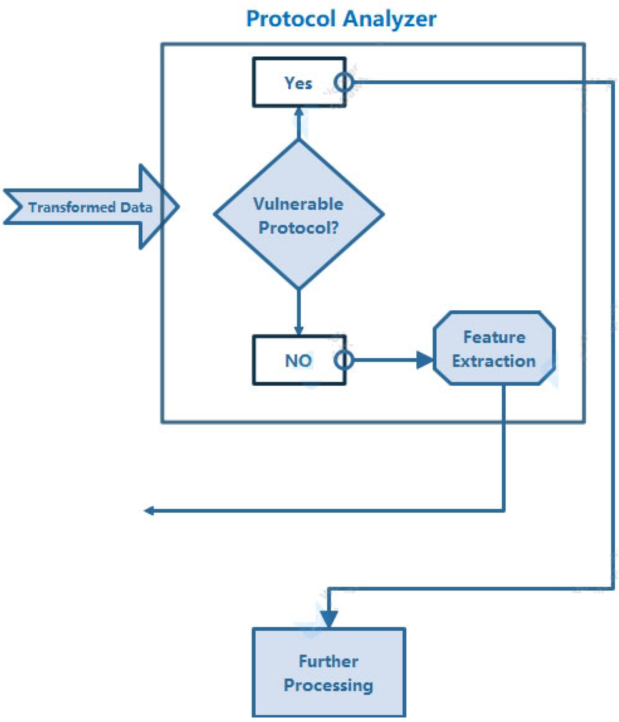


Figure 8: Protocol Analyzer

3.4 Feature selection

The datasets [11] involve different features that are often classified into below groups:

1. Flow features: this group includes the identifier attributes between hosts, such as client-to-server or server-to-client
2. Basic features: this category involves the attributes that represent protocols connections
3. Content features: this group encapsulates the attributes of TCP/IP; also, they contain some attributes of HTTP services.
4. Time features: this category contains the attributes of time, for example, arrival time between packets, start or end packet time and round-trip time of TCP protocol.
5. Additional generated features: this category can be further divided into two groups:
   (a) General purpose features where each feature has its own purpose, in order to protect the service of protocols.
   (b) Connection features are built from the flow of 100 record connections based on the sequential order of the last time feature.
6. Labelled Features: this group represents the label of each record

However network packets carry a wide variety of irrelevant or redundant feature, we use feature selection to examine our dataset and remove the unwanted features that affect the efficiency and detection rate of our algorithms. For this purpose, we applied two feature selection models which are Chi-2 and RFE to find the best features from the dataset. the are defined as follows:

- Chi-2:Chi square measures the dependency between a feature and a class by counting the occurrence of the feature with respect to occurrence of the class. Chi2 is simple but effective if a feature with a certain distribution can be differentiated easily in normal and attack packets. In this method, features with highest scores are selected.

- RFE:This method first calculates the importance of each feature from a full features list based on a trained estimator, which can be a simple machine learning algorithm. Then, RFE removes features having the least importance value from the subset recursively until a desired length of feature list is reached.

3.5 Proposed Algorithms

For performance testing, the selected features are applied to different algorithms including Multi-Layer Perceptron (MLP), SVM, k-Nearest Neighbor (k-NN), Decision Tree (DT) and Logistic Regression (LR). The best algorithms were selected through a benchmark on applied datasets and comparing the results using metrics like accuracy, False Positives (FPs), False Negatives (FNs), training and testing time. The applied algorithms are described below.

- MLP: A multilayer perceptron (MLP) is a perceptron that teams up with additional perceptrons, stacked in several layers, to solve complex problems. The diagram below shows an MLP with three layers. Each perceptron in the first layer on the left (the input layer), sends outputs to all the perceptrons in the second layer (the hidden layer), and all perceptrons in the second layer send outputs to the final layer on the right (the output layer).[5] sends multiple signals, one signal going to each perceptron in the next layer. For each signal, the perceptron uses different weights. In the diagram above, every line going from a perceptron in one layer to the next layer represents a different output. Each layer can have a large number of perceptrons, and there can be multiple layers, so the multilayer perceptron can quickly become a very complex system. The multilayer perceptron has another, more common name—a neural network. A three-layer MLP, like the diagram above, is called a Non-Deep or Shallow Neural Network. An MLP with four

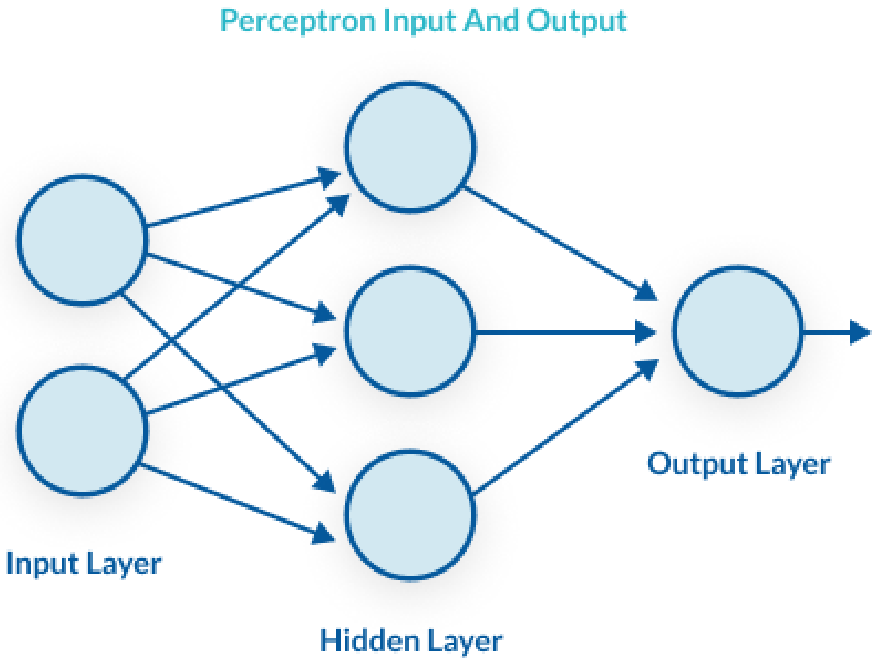


Figure 9: multilayer perceptron algorthm.[5]

or more layers is called a Deep Neural Network. One difference between an MLP and a neural network is that in the classic perceptron, the decision function is a step function and the output is binary. In neural networks that evolved from MLPs, other activation functions can be used which result in outputs of real values, usually between 0 and 1 or between -1 and 1. This allows for probability-based predictions or classification of items into multiple labels.

- SVM: Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

  The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:[6]

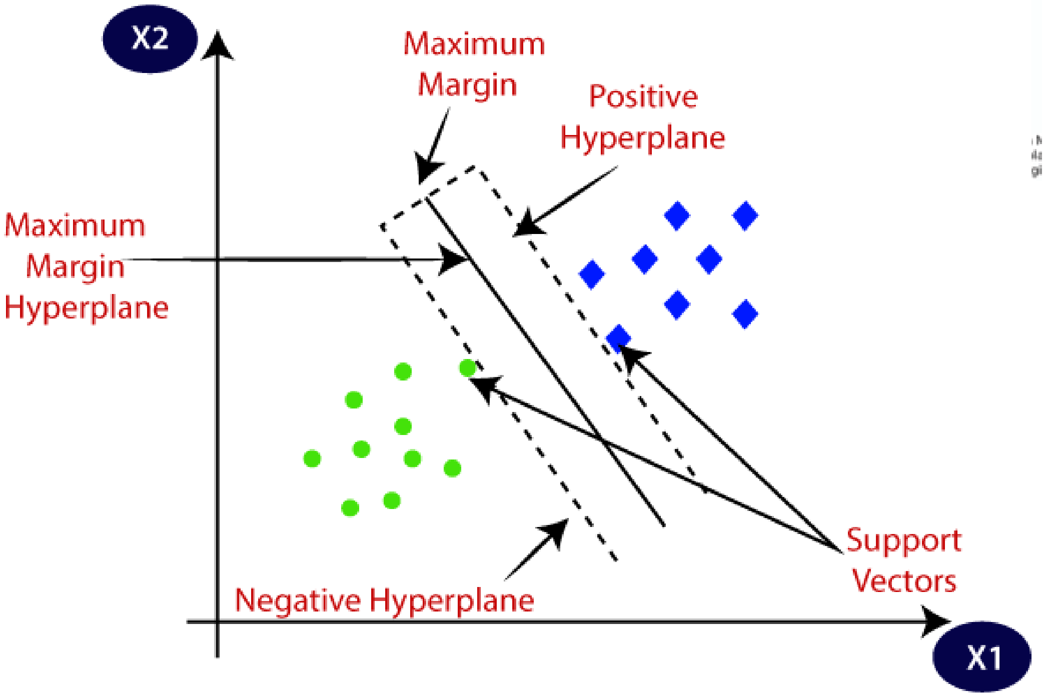


Figure 10: support vector machine algorthm.[6]

- KNN: K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.[7]

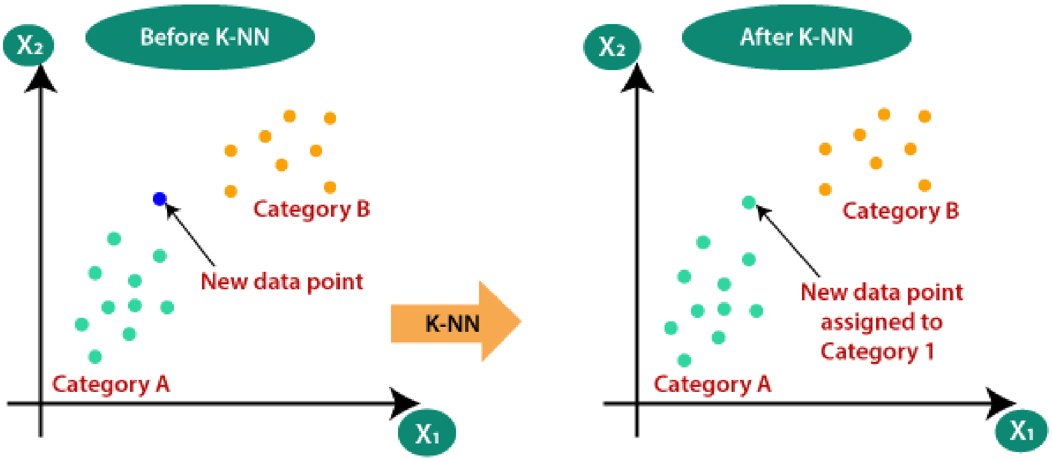


Figure 11: k nearest neighbor algorithm for machine learning.[7]

- DT:Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees. Below diagram explains the general structure of a decision tree:[8] The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

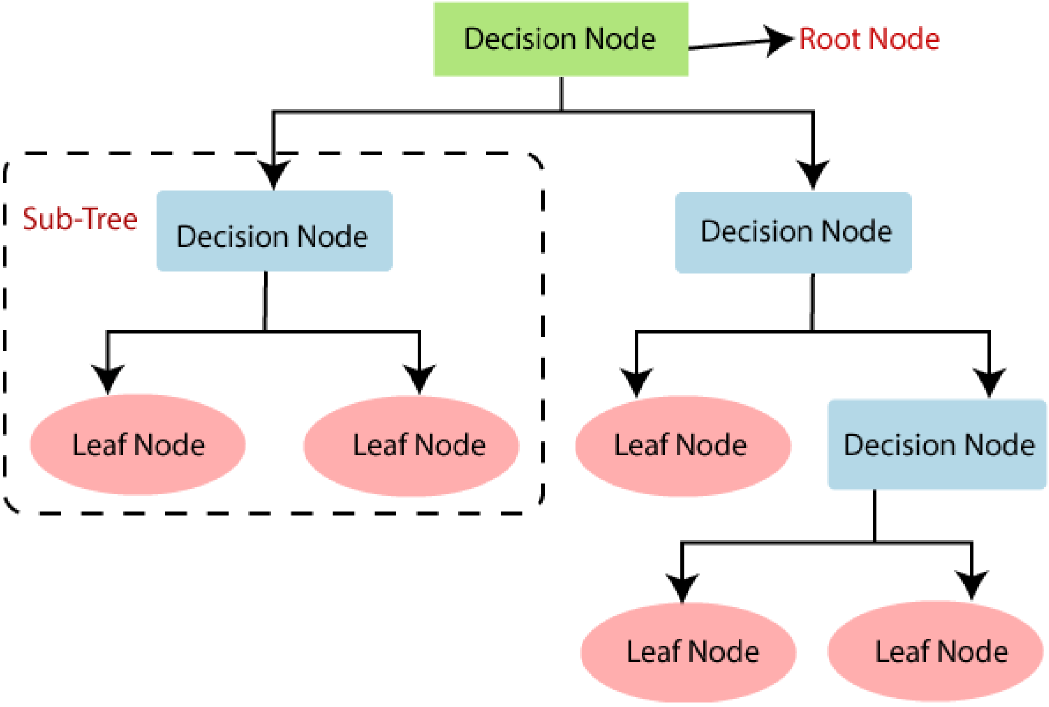


Figure 12: decision tree classification algorithm.[8]

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

- LR:Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:[9]

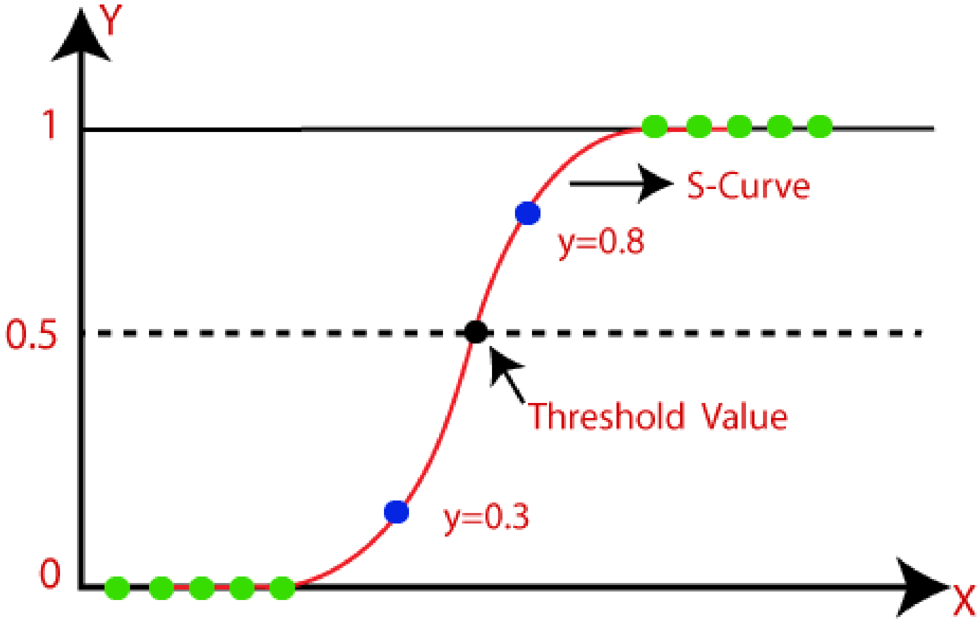


Figure 13: logistic regression algorithm in machine learning.[9]

4 Results and Discussion

4.1 Development Environment

The research was done using the Google Colab platform with a CPU Architecture of x86, Intel(R) Xeon(R) CPU@ 2.30GHz, two core and two Threads per core. The Ram capacity was 12 GB upgradable to 26.75GB. All performed on a windows 10 Computer. The dataset was encoded and normalize using the LabelEncoder and MinMaxScaler from the Sklearn library. On the other hand, the graphs were plotted using the Matplotlib and the Seaborn libraries.

Parameter	**Value**
Environment	Google Colab
Langage	Python 3.6.9
CPU Model name	2.30GHZ
No.CPU Cores	2
CPU Family	Haswell
RAM	12GB (upgradable to 26.75GB)
Disk Space	25GB

Table 1: Specifications of the development environment

4.2 Dataset Description

We used two publicly available datasets namely the UNSW-NB15 and the NSL-KDD. The UNSW-NB15 dataset's raw network packets were generated by the IXIA PerfectStorm tool in the Australian Centre for Cyber Security's (ACCS) Cyber Range Lab to create a hybrid of real modern normal activities and synthetic contemporary attack behaviors[24]. In this dataset, attacks are grouped in to nine categories: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. It has a total of 49 features with the class label. The data is divided in training and testing set with 175,341 records and 82,332 records respectively for each.
On the other hand, NSL-KDD dataset is an update of the KDD'99 that suffered from redundancy and duplication[25]. It has a total of 43 features and four different classes of attacks: Denial of Service (DoS), Probe, User to Root(U2R), and Remote to Local (R2L). The rest of the dataset description is shown in the graphs below.

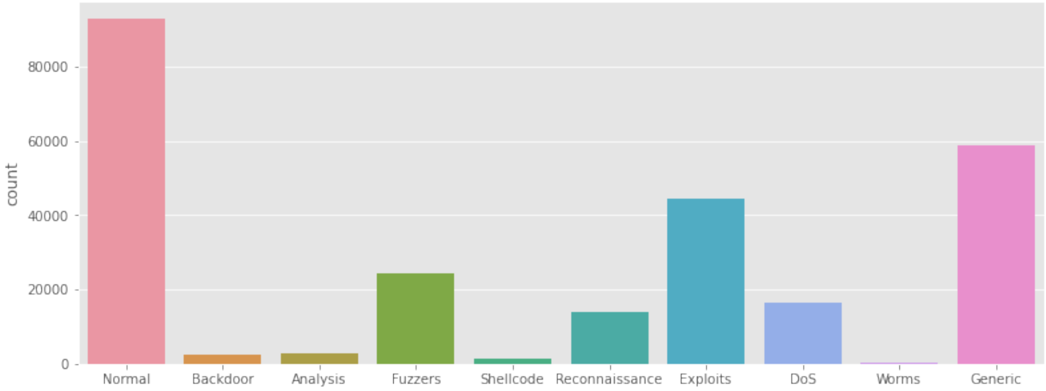


Figure 14: UNSW-NB15 Contamination Ratio

Both datasets had the same contamination ratio. The number of packets labelled as normal versus packets labelled as an attack is shown in Figure 15.

The protocol being an important feature on which our model is based, we have split the dataset as indicated in the protocol Analyzer. The number of records per-protocol is shown in Figure 16.

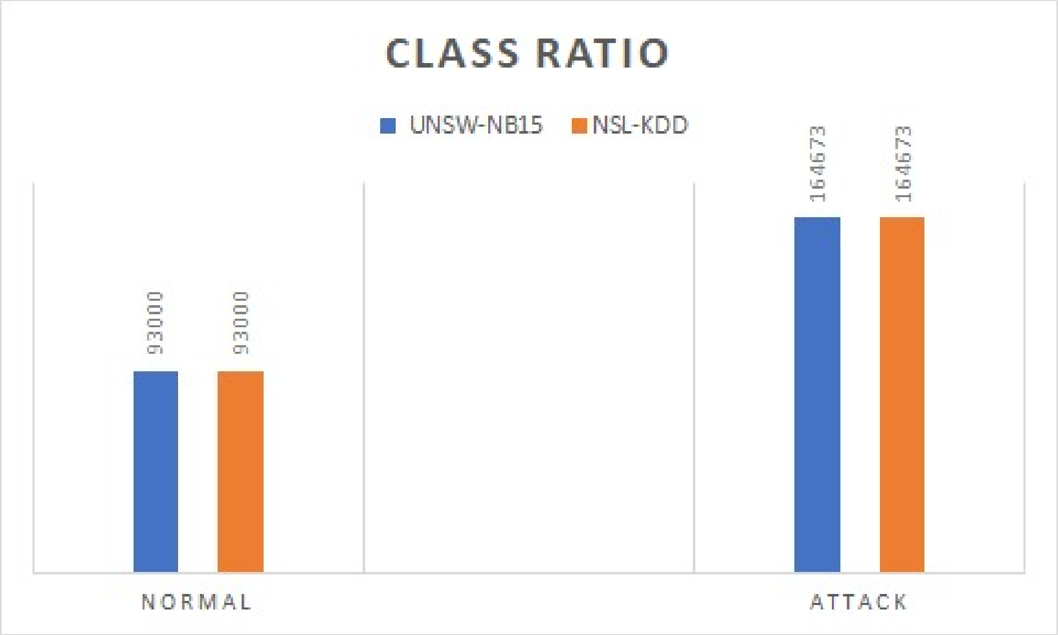


Figure 15: UNSW-NB15 Class Ratio

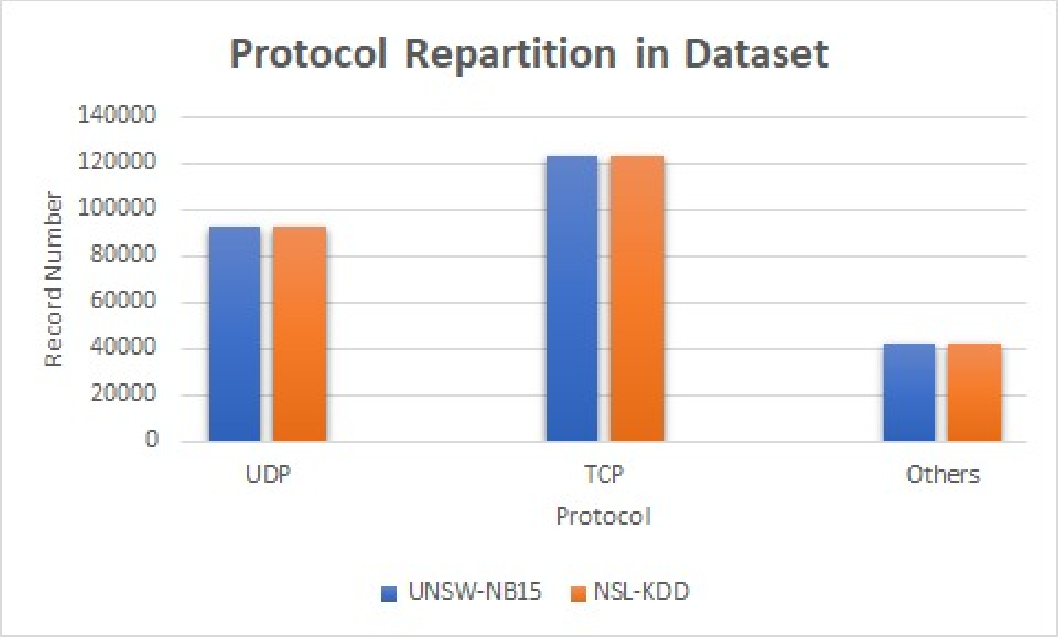


Figure 16: Number of Records Per Protocol

After the protocol Analyzer step in which traffics were divided according to their protocol, we applied the Chi2 and RFE feature selection. To train each model, we selected the 25 best features using the above mentioned feature selection techniques. The rankings of the features based on the score computed by each technique are shown in Figure 17 and Figure 18. The choice of selecting 25 features was based on the fact that a number below this was not leading to better performance of various models. Reducing the function to use only the 25 most important features reduced the training time, Overfitting and improved the accuracy.

The Traffics using UDP Protocol are sent to the candidate algorithm KNN while those using TCP protocol are sent to LR algorithm. For the rest of the traffic using other protocols than UDP and TCP, we applied the MLP algorithm.

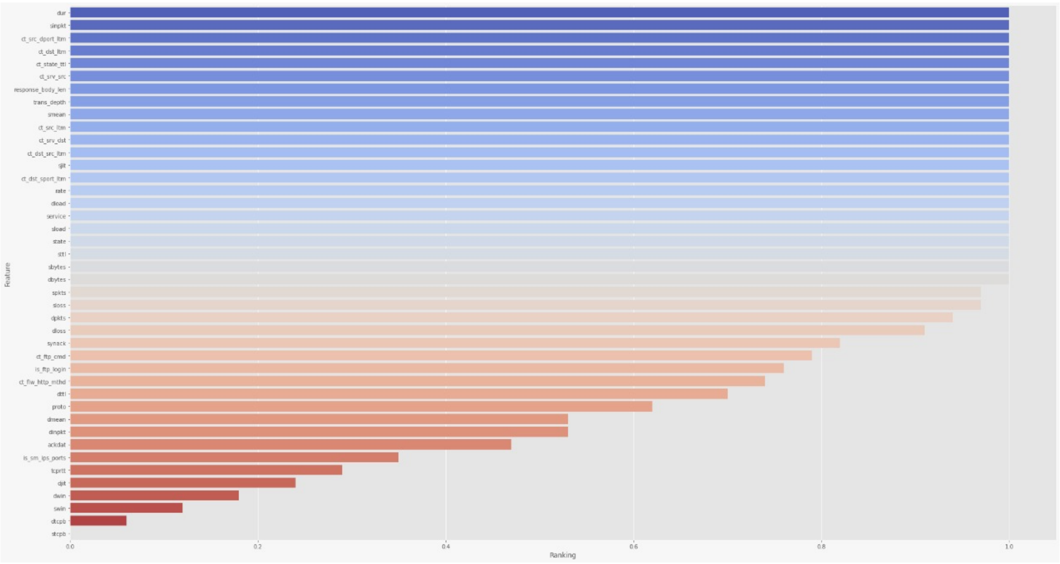


Figure 17: Feature ranking by CHI2

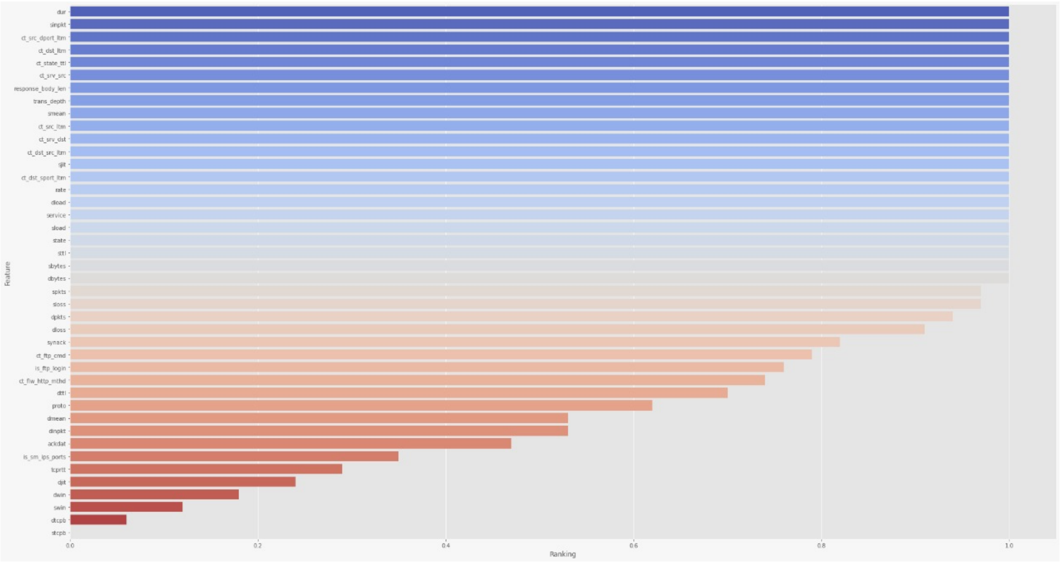
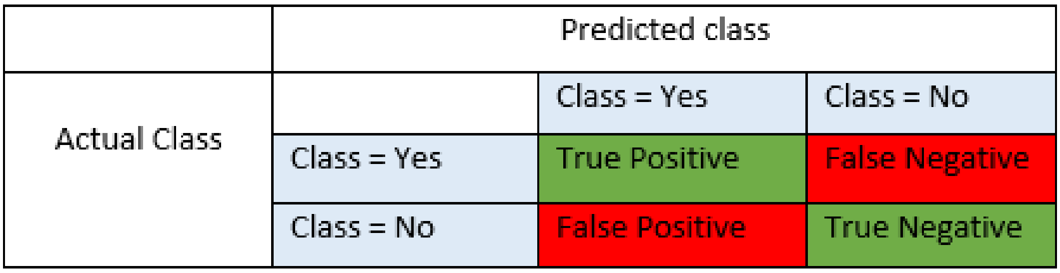


Figure 18: Feature ranking by RFE

4.3 Performance metrics

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. All the measures except AUC can be calculated by using left most four parameters. [10] So, let's talk about those four parameters first.



		Predicted class	
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Figure 19: Confusion Matrix.[10]

True positive and true negatives are the observations that are correctly predicted and therefore shown in green. We want to minimize false positives and false negatives so they are shown in red color. These terms are a bit confusing. So let's take each term one by one and understand it fully.

4.3.1 True Positives (TP)

- These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

4.3.2 True Negatives (TN)

- These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

False positives and false negatives, these values occur when your actual class contradicts with the predicted class.

4.3.3 False Positives (FP)

– When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

4.3.4 False Negatives (FN)

– When actual class is yes but predicted class is no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

Once we understand these four parameters then we can calculate Accuracy, Precision, Recall and F1 score.

4.3.5 Accuracy

- Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of our model[10].

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

4.3.6 Precision

- Precision helps when the costs of false positives are high. So let's assume the problem involves the detection of skin cancer. If we have a model that has very low precision, then many patients will be told that they have melanoma, and that will include some misdiagnoses. Lots of extra tests and stress are at stake. When false positives are too high, those who monitor the results will learn to ignore them after being bombarded with false alarms.

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

4.3.7 Recall (Sensitivity)

- Recall is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made. Unlike precision that only comments on the correct positive predictions out of all positive predictions, recall provides an indication of missed positive predictions. In this way, recall provides some notion of the coverage of the positive class.

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

The result is a value between 0.0 for no recall and 1.0 for full or perfect recall.

4.3.8 F1 score

- F1 is an overall measure of a model's accuracy that combines precision and recall, in that weird way that addition and multiplication just mix two ingredients to make a separate dish altogether. That is, a good F1 score means that you have low false positives and low false negatives, so you're correctly identifying real threats and you are not disturbed by false alarms. An F1 score is considered perfect when it's 1, while the model is a total failure when it's 0. [26]

$$F1Score = 2 * \frac{(Recall * Precision)}{(Recall + Precision)}$$

4.4 Discussion

Table 2 represents the performance measurement of all the candidate algorithms. The KNN, LR and MLP were trained on the training set and their accuracy, precision, F-1 score, and Recall were measured respectively on the test set using the feature transformation (FT) proposed by our approach and without using it. As shown in table 2, for each algorithm, we applied the previously mentioned feature selection separately and measured the performances.
For UDP traffics, our proposed method of using KNN combined with the feature transformation showed a significant improvement. The combination outperformed the simple KNN when using the RFE selection with an accuracy of 0.95 against 0.63 which is almost an increase of 50%. Similarly, the F-1 score and the Recall increased by 26% and 50% respectively.

For the TCP traffics, the combination of the proposed feature selection with the LR algorithm using RFE selection increased the accuracy from 0.55 to 0.67 which is an improvement of around 21%. Likewise, the precision, the F-1 score, and the recall rose by 140%, 69% and 21% respectively.

For the other protocols, a similar improvement was observed with the MLP combined to FT using RFE that showed an increase in accuracy of 19%. The Precision, the F-1 score. and the Recall also showed an improvement of 6%, 11%, and 16% respectively.

Model Name	Feature Selection	Accuracy	Precision	F-1 Score	Recall
KNN	RFE	0.63	0.94	0.73	0.63
	CHi2	0.05	0.00	0.01	0.05
KNN + FT	RFE	0.95	0.90	0.92	0.95
	CHi2	0.36	0.84	0.50	0.36
LR	RFE	0.55	0.30	0.39	0.55
	CHi2	0.80	0.85	0.80	0.80
LR + FT	RFE	0.67	0.72	0.66	0.67
	CHi2	0.46	0.75	0.30	0.46
MLP	RFE	0.84	0.94	0.90	0.86
	CHi2	1.00	1.00	1.00	1.00
MLP + FT	RFE	1.00	1.00	1.00	1.00
	CHi2	1.00	1.00	1.00	1.00

Table 2: Performance measurements of candidate algorithms(weighted avg)

The experiments showed that the feature transformation technique can perform better while combined with the RFE rather than CHi2. This can be explained by the fact that the naive Bayes feature embedding that we proposed magnified the difference between the normal and the attack, which makes it easy to separate normal packet from attacks. Also, the computation time considerably reduced notably due to our protocol Analyzer that filtered vulnerable protocols such as TCP and UDP to avoid unnecessary computation overhead. The overall System performed well with good accuracy compared to the traditional model in which the protocol Analyzer and the feature transformation were not used. This is summarized in Figure 20

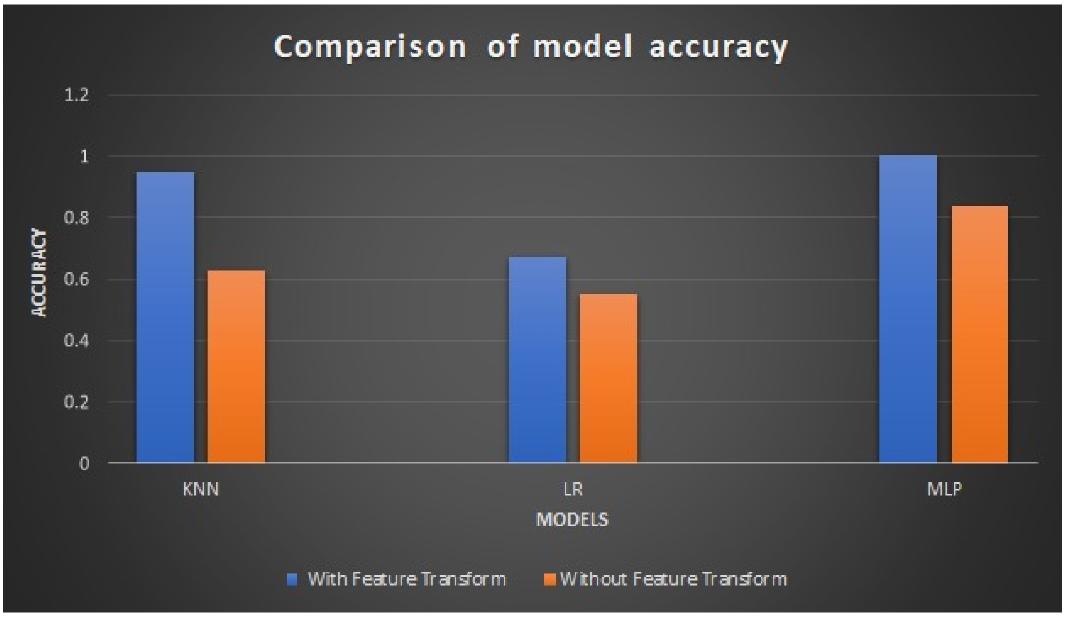


Figure 20: Accuracy Comparison of Various Models

Figure 21 shows the true and false-positive prediction of our proposed approach against the traditional approach. The KNN model produced the highest prediction of the true positive when using our FT the same thing goes for the LR and MLP models.

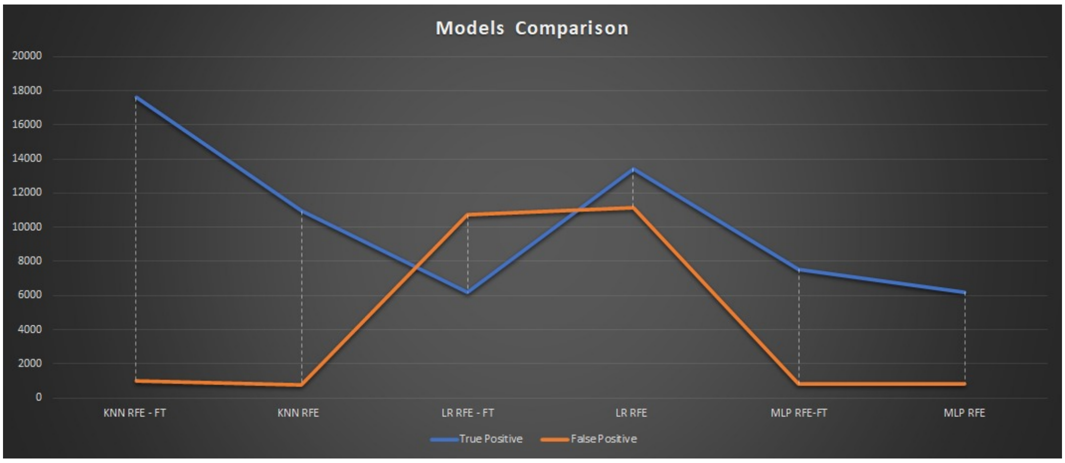


Figure 21: Positive rate of Candidate algorithms

5 Conclusion and Future Work

To conclude, we have built a hybrid intrusion detection system capable of detecting previously seen and unseen attacks. This model uses a combination of Feature transformation and protocol analyzer to improve data quality. This newly generated data is much more concise, informative and take lest time for fitting our candidate algorithms and considerably reduces the processing time. We evaluated our approach using various performance metrics such as accuracy, F-1 score, precision, and Recall. The outcome of this evaluation proved that our proposed method can perform better than the traditional approach used with the simple candidate algorithms.

During the development process, we faced some challenges such as the huge number of records in the dataset and the unbalanced class distribution. To solve this we had to use under sampling and over sampling techniques proposed by the Scikit-learn library.

In the future, as our actual work is based on just detecting if a data packet is an attack or a normal packet, we wish to perform attacks classification based on their type. On the other hand, we will add a misuse detection module that can profile users behaviours. This misuse detection can improve the detection rate by learning request frequencies sent by users and various IoT sensors.

References

[1] S. G. Kene and D. P. Theng, "A review on intrusion detection techniques for cloud computing and security challenges," in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pp. 227–232, 2015.

[2] K. Khan, S. Khan, M. Altaf, Z. Iqbal, and W. Mashwani, "A survey on intrusion detection and prevention in wireless ad-hoc networks," *Journal of Systems Architecture*, vol. 105, p. 101701, 12 2019.

[3] N. T. Van, H. Bao, and T. N. Thinh, "An anomaly-based intrusion detection architecture integrated on openflow switch," in *Proceedings of the 6th International Conference on Communication and Network Security*, ICCNS '16, (New York, NY, USA), p. 99–103, Association for Computing Machinery, 2016.

[4] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks," *Electronics*, vol. 8, no. 11, 2019.

[5] Missinglink.ai, "Perceptrons and multi-layer perceptrons: The artificial neuron at the core of deep learning."

[6] J. T. Point, "Support vector machine algorithm."

[7] J. T. Point, "K-nearest neighbor(knn) algorithm for machine learning."

[8] J. T. Point, "Decision tree classification algorithm."

[9] J. T. Point, "Logistic regression in machine learning."

[10] C. Nicholson, "Accuracy, precision, recall f1 score: Interpretation of performance measures." `https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/`, 2016.

[11] M. Monshizadeh, V. Khatri, B. G. Atli, R. Kantola, and Z. Yan, "Performance evaluation of a combined anomaly detection platform," *IEEE Access*, vol. 7, pp. 100964–100978, 2019.

[12] D. S. Robert P. Markiewicz, "Clust-it: Clustering-based intrusion detection in iot environments," 2020.

[13] D. Zheng, Z. Hong, N. Wang, and P. Chen, "An improved lda-based elm classification for intrusion detection algorithm in iot application," *Sensors*, vol. 20, no. 6, p. 1706, 2020.

[14] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban ids: An intelligent anomaly based intrusion detection system for iot edge devices," *IEEE Internet of Things Journal*, 2020.

[15] J. O.-G. A. L. M.-C. I. G. Hector Alaiz-Moreton, Jose Aveleira-Mata and C. Benavides, "Multiclass classification procedure for detecting attacks on mqtt-iot protocol," 2019.

[16] C. G. Tianqi Chen, "Xgboost: A scalable tree boosting system," 2016.

[17] C. G. e. a. K. Cho, B. van Merrienboer, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[18] M. A. F. A. Zaffar Haider Janjua, Massimo Vecchio, "Irese: An intelligent rare-event detection system using unsupervised learning on the iot edge," 2019.

[19] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, and K.-K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks," *IEEE Transactions on Emerging Topics in Computing*, 2016.

[20] A. Mudgerikar, P. Sharma, and E. Bertino, "E-spion: A system-level intrusion detection system for iot devices," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pp. 493–500, 2019.

[21] M. Niedermaier, M. Striegel, F. Sauer, D. Merli, and G. Sigl, "Efficient intrusion detection on low-performance industrial iot edge node devices," *arXiv preprint arXiv:1908.03964*, 2019.

[22] F. Alghayadh and D. Debnath, "A hybrid intrusion detection system for smart home security based on machine learning and user behavior," *Advances in Internet of Things*, vol. 11, no. 1, pp. 10–25, 2021.

[23] J. Gu and S. Lu, "An effective intrusion detection approach using svm with naïve bayes feature embedding," *Computers & Security*, p. 102158, 2020.

[24] N. M. J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," 2015.

[25] M. T. E. B. W. L. A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," 2009.

[26] C. Nicholson, "Evaluation metrics for machine learning - accuracy, precision, recall, and f1 defined."