INTRUSION DETECTION SYSTEM BASED ON MACHINE LEARNING

A PROJECT REPORT

Submitted by

PRIYA P R (TKM19MCA018)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

MAY 2022

DECLARATION

I undersigned hereby declare that the project report INTRUSION DETECTION SYSTEM BASED ON MACHINE LEARNING , submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Alshaina S. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam

25/05/2022

PRIYA P R

Thangal Kunju Musaliar College of Engineering

Dept. of Computer Applications

C E R T I F I C A T E

This is to certify that, this report titled ***Intrusion Detection System Based On Machine Learning*** is a bonafide record of the **Project** presented by **PRIYA P R (TKM19MCA018)**, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, **MCA Computer Application** in **APJ Abdul Kalam Technological University**.

Internal Supervisor | Head of the Department | External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of our project.

I am extremely grateful to **Dr.Fousia M Shamsudeen**, Head of the Department, for providing us with best facilities.

I would like to thank my project guide **Prof.Alshaina S**, Department of Computer Applications, who motivated me throughout the project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout our course of study.

I owe my thanks to our friends and all others who have directly or indirectly helped us in the successful completion of this project.

Priya P R

Abstract

Intrusion Detection System is a software application to detect network intrusion. Network Intrusion Detection System(NIDS) needs to accurately identify malicious network attacks. Network intrusion using various machine learning algorithms to helps to detect network intrusion. In imbalanced network traffic, malicious cyber attacks can often hide in large amounts of normal data. It exhibits a high degree of stealth and obfuscation in cyberspace, making it difficult for Network Intrusion Detection System(NIDS) to ensure the accuracy and timeliness of detection. It proposes a novel Difficult Set Sampling Technique(DSSTE) algorithm to tackle the class imbalance problem. First, use the Edited Nearest Neighbor(ENN) algorithm to divide the imbalanced training set into the difficult set and the easy set. Next, use the KMeans algorithm to compress the majority samples in the difficult set to reduce the majority. Zoom in and out the minority samples' continuous attributes in the difficult set synthesize new samples to increase the minority number. Finally, the easy set, the compressed set of majority in the difficult, and the minority in the difficult set are combined with its augmentation samples to make up a new training set.The algorithm reduces the imbalance of the original training set and provides targeted data augment for the minority class that needs to learn. It enables the classifier to learn the differences in the training stage better and improve classification performance. The classic intrusion dataset CSE-CIC-IDS2018 is used to verify the proposed method. The proposed method use Logistic Regression Model, Decision Tree Classifier Model , K Nearest Neighbors Classifier Model and Naive Bayes model. Compare this algorithm to identify which perform better in intrusion detection.

Contents

List of Figures

List of Tables

List of Abbreviations

NIDS	Network Intrusion Detection System
DSSTE	Difficult Set Sampling Technique
CSE CIC	Communications Security Establishment the Canadian Institute for Cybersecurity
KNN	K Nearest Neighbor
ENN	Edited Nearest Neighbor

Chapter 1

Introduction

With the rapid development and wide application of 5G, IoT, Cloud Computing, and other technologies, network scale, and real-time traffic become more complex and massive, cyber-attacks have also become complex and diverse, bringing significant challenges to cyberspace security. As the second line of defense behind the firewall, the Network Intrusion Detection System(NIDS) needs to accurately identify malicious network attacks, provide real-time monitoring and dynamic protection measures, and formulate strategies. In real cyberspace, normal activities occupy the dominant position, so most traffic data are normal traffic; only a few are malicious cyber attacks, resulting in a high imbalance of categories. In the highly imbalanced and redundant network traffic data, intrusion detection is facing tremendous pressure. Cyber attacks can hide in a large amount of normal traffic.

Faced with imbalanced network traffic data, The proposed system use novel Difficult Set Sampling Technique(DSSTE) algorithm to tackle the class imbalance problem in network traffic. This method effectively reduces the imbalance

and makes the classification model learning difficult samples more effective. In this system use classic machine learning algorithms to verify a benchmark data sets. The specific contributions are as follows.

(1) The up-to-date CSE CIC-IDS2018 as benchmark data sets and conduct detailed analysis and data cleaning.

(2) This work proposes a novel DSSTE algorithm, reducing the majority samples and augmenting the minority samples in the difficult set, tackling the class imbalance problem in intrusion detection so that the classifier learns the differences better in training.

(3) The classification model uses Logistic Regression Model, Decision Tree Classifier Model , K Nearest Neighbors Classifier Model and Naive Bayes model.

1.1 Problem Definition

The proposed system utilize different machine learning algorithm to predict intrusion in imbalanced network traffic. Then compare this algorithm to identify which perform better in intrusion detection.

1.2 Objectives

Intrusion Detection system is a detective device designed to detect malicious attack. IDS are generally deployed with the purpose to monitor and analyze user and system activity. In this work, feature selection is carried out with Random Forest. The selected features are then input to classification models like Logistic Regression Model, Decision Tree Classifier Model , K Nearest Neighbors Classifier Model and Naïve Bayes Classifier model. The performance comparison is done to identify the best model.

Chapter 2

Literature Survey

Literature review is the comprehensive study and interpretation of literature that relates to a particular topic. When one uses literature review research questions are identified, then one seek to answer this research questions by searching for and analyzing relevant literature. Some importance of literature reviews is that new insights can be developed by the re-analyzing the results of the study. A literature review is both a summary and explanation of the complete and current state of knowledge on a topic as found in academic books and journal articles. There are two kinds of literature reviews you might write at university: one that students are asked to write as a stand-alone assignment in a course, and the other that is written as part of an introduction to, or preparation for, a longer work, usually a thesis or research report. The focus and perspective of your review and the kind of hypothesis or thesis argument you make will be determined by what kind of review you are writing. One way to understand the differences between these two typesis to read published literature reviews or the first chapters of theses and dissertations in your own subject area. Analyses the structure of their arguments and note the way they address the issues.

2.1 Purpose of the Literature Review

1. It gives readers easy access to research on a particular topic by selecting high quality articles or studies that are relevant, meaningful, important and valid and summarizing them into one complete report.

2. It provides an excellent starting point for researchers beginning to do research in a new area by forcing them to summarize, evaluate, and compare original research in that specific area.

3. It ensures that researchers do not duplicate work that has already been done.

4. It can provide clues as to where future research is heading or recommend areas on which to focus.

5. It highlights the key findings.

6. It identifies inconsistencies, gaps and contradictions in the literature.

7. It provides a constructive analysis of the methodologies and approaches of other researchers.

2.2 Related Works

The section mainly describing what all are the related works that have been occurred in Intrusion detection system . And some of them are listed below.

Bhattacharya et al.[1] Proposed a machine learning model based on hybrid Principal Component Analysis(PCA)-Firefly. The dataset used was the open dataset collected from Kaggle. Firstly, the model performs one key coding for transforming the IDS dataset,then uses the hybrid PCA-Firefly algorithm to reduce the dimension, and the XGBoost algorithm classifies the reduced dataset.In recent years, with the powerful ability of automatic feature extraction, deep learning has made remarkable achievements in the fields of Computer Vision(CV), Autonomous driving(AD), Natural Language Processing(NLP).

P. Bedi, N. Gupta and V. Jindal et al.[2] Proposed a new method for feature selection and classification merging of multi-class NSL-KDD Cup99 dataset using Support Vector Machine(SVM) and discussed the classification accuracy of classifiers under different dimension features.

G. Caminero, M. Lopez-Martin, and B. Carro et al. [3] Studied some new technologies to improve CANN intrusion detection methods'classification performance and evaluated their performanceon the NSL-KDD Cup99 dataset.

A. K. Verma et al. [4] proposed an improved local adaptive composite minority sampling algorithm (LA-SMOTE) to deal with the network traffic imbalance problem and then based on the deep learning GRU neural network to detect the network traffic anomaly.

B. A. Tama, M. Comuzzi, and K.-H. Rhee et al. [5] He used the K Farthest Neighbor(KFN) and the K Nearest Neighbor(KNN) to classify the data and used the Second Nearest Neighbor(SNN) of the data when the nearest and farthest neighbors have the same class label. The result shows the CANN detection rate and reduces the failure the alert rate is improved or provides the same performance.

A. Raghavan et al. [6] proposed an anomaly-based IDS based on a two-stage meta-classifier, which uses a hybrid feature selection method to obtain accurate feature representations.They conducted on the proposed method on the NSL-KDD and UNSW-NB15 intrusion datasets and improved detection rates.

J.-T. Wang and C.-H. Wang et al. [7] this method combines the Synthetic Minority Over sampling Technique(SMOTE) and Complementary Neural Network(CMTNN) to solve imbalanced data classification. Experiments on the UCI dataset show that the proposed combination technique can improve class imbalance problems.

D. Kwon et al. [8] Apply deep learning to intrusion detection for traffic classification, which has become a hot spot of current research. The method of deep learning is to mine the potential characteristics of high-dimensional data through a training model and transform network traffic anomaly detection into classification problem.

R. Abdulhammed et al. [9] By comparing with the traditional machine learning technology, it is proved that the FCN model is useful for network traffic analysis.

A. Ismail et al. [10] proposed an intrusion detection algorithm based on Long Short-Term Memory(LSTM), which detects DoS attacks and probe attacks with unique time series in the KDD Cup99 dataset.

N.Shone et al. [11] first converted network traffic characteristics into a series of characters and then used Recurrent Neural Network(RNN) to learn their temporal characteristics,which were further used to detect malicious network traffic.

T. Young et al. [12] proposed a malicious software traffic classification algorithm based on Convolutional Neural Network(CNN). By mapping the traffic character-

istics to pixels, the network traffic image is generated, and the image is used as the input of the CNN to realize traffic classification.

X. Lei and Y. Xie et al. [13] has carried out relevant research on the deep learning model, focusing on data simplification, dimension reduction, classification, and other technologies, and proposes a Fully Convolutional Network(FCN) model.

B. Yan and G. Han et al.[14] Through a large number of sample data training, adaptive learning between normal network traffic and abnormal network traffic effectively enhances real-time intrusion processing.

I. Sharafaldin et al. [15] deal with the imbalanced dataset CIDDS001 using data Up sampling and Down sampling methods, and by Deep Neural Networks, Random Forest, Voting, Variational Autoencoder, and Stacking Machine Learning classifiers to evaluate datasets.

H. Shapoorifard et al [16] first converted network traffic characteristics into a series of characters and then used Recurrent Neural Network(RNN) to learn their temporal characteristics, which were further used to detect malicious network traffic.

X. Ma and W. Shi et al. [17] proposed a malicious software traffic classification algorithm based on Convolutional Neural Network(CNN). By mapping the traffic characteristics to pixels, the network traffic image is generated, and the image is used as the input of the CNN to realize traffic classification.

M. Lopez-Martin et al. [18] has carried out relevant research on the deep learning model, focusing on data simplification, dimension reduction, classification, and other technologies, and proposes a Fully Convolutional Network(FCN) model. By comparing with the traditional machine learning technology, it is proved that the FCN model is useful for network traffic analysis.

Chapter 3

Intrusion Detection System

Intrusion Detection System (IDS) is a system which monitors the pattern of network flow and helps in distinguishing normal traffic and malicious traffic. An IDS system can be deployed in small scale systems or large scale systems i.e., it's scope can range from single computers to large networks of computers. As it's name suggest this is used to detect different attacks. Attack can be either host-based or network-based. Host based attacks occur in a single system. These can be either in the form of worms, viruses, trojans or backdoor. To prevent these attacks Host-based IDS is developed. They are deployed within the host and they monitor the network flow which originated from a particular host only. These systems also analyze the file systems, login/off activities, data processing and so on. Network-based attacks occur in an inter-connected network. These can be either a DDOS attack, IP spoofing, Port sniffing or Man-in-the-Middle attack. Now to mitigate such attacks Network-based IDS is developed. Fig 3.1 shows different types of IDS. It is positioned in the network in such a way that any attack in the host connected to that particular network will be detected i.e., it will be placed in the entry and exit point of data from that network.

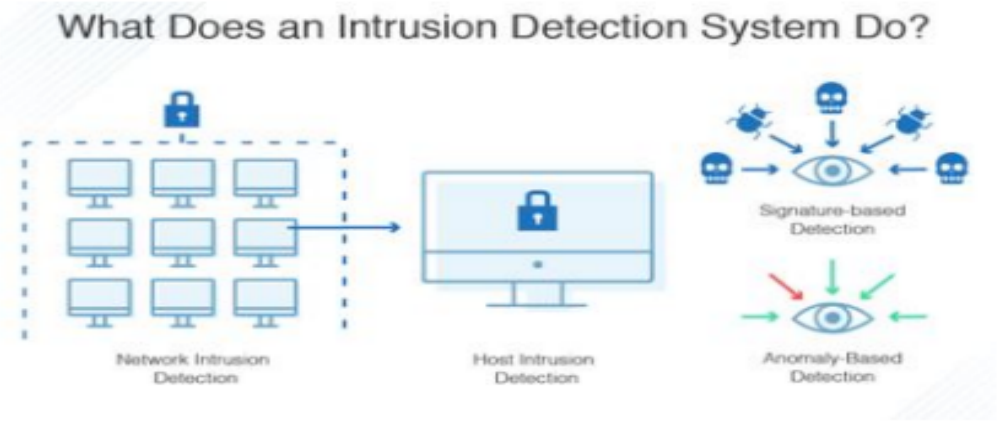

Figure 3.1: Types of IDS

3.1 Challenges of IDS

All these types of IDS can either be signature-based or anomaly-based. Fig 3.2 shows the flowchart of different IDS techniques. Signature-based IDS otherwise known as misuse-based IDS system are developed in such a way that any pre-defined patterns of network traffic will be easily identified whereas Anomaly-based IDS detects attacks based on variations from the normal instances which are already defined. The main disadvantage of signature-based system is that they will not be able to identify unseen attacks. But deployment of both the techniques were not able to reduce the problem of false alarms i.e., all these techniques are based on strict rules and they are vulnerable to false positive and false negative alarms. Apart from that the computing cost is high as the network traffic types is increasing day by day and the network characteristics are becoming complex as the attackers are changing the features periodically.

3.2 Network Intrusion Detection System

Conventional Network IDS works by building statistical tools or models around the network traffic that they monitor as shown in fig 3.3. The current network traffic flow will be captured and this data will be send to the server. The NIDS server will then process the data to detect the threat. It will detect a threat whenever there is a significant deviation from the baseline models and once such threat is detected it will raise alerts in the form of alarms. The main issue with conventional NIDS is the high rate of false-alarm rates which reduces the detection rate. As part of improving the system the researches started to work on NIDS models. Advancement in machine learning techniques paved way for making these conventional models intelligent. Thus intelligent NIDS were developed with shallow machine learning techniques like SVM, Naive Bayes, Logistic Regression, Decision tree, Random Forest and so on. Several deep learning techniques were also employed to improve the detection rate and reduce the false alarm rate. Now the development has reached to the extend of fusing machine learning and deep learning approaches.

Intrusion Detection System

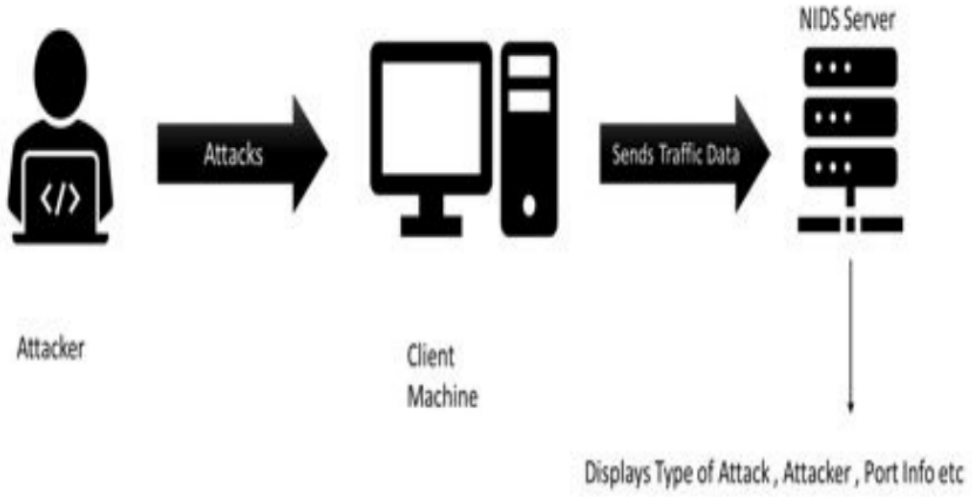

Figure 3.1: Intrusion Detection System

Chapter 4

Methodology

Data preprocessing, parameter tuning, feature extraction and classification are the four important steps for the accomplishment of this work. The algorithm includes:

-

1. Data preprocessing and normalization for modelling different scales.
2. Perform parameter tuning to assign the best values to the model.
3. Apply features to the classification model.
4. Compare their performance and evaluate the False Alarm Rate.

This work proposed the intrusion detection model as shown in figure 4.1.The proposed system utilized DSSTE algorithm to convert imbalanced data in to balanced data. Firstly data preprocessing is performed in the intrusion detection structure, for removing duplicate outliers and missing value.Then partioning the test set and training set. The training set is processed for data balancing using DSSTE algorithm. This training set is divided into near neighbor and far neighbor set by edited nearest neighbor algorithm. The sample in nearest neighbor set are considered as difficult set and far neighbor set as easy set. Difficult set are again classified in

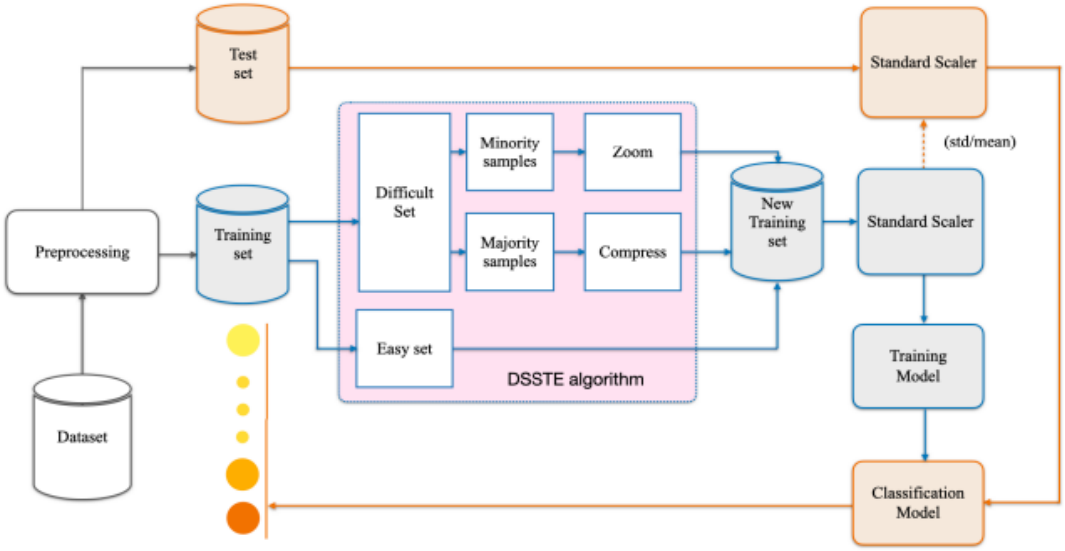

Figure 4.1: Block diagram for intrusion detection System

minority sample and majority sample. DSSTE algorithm compress the majority sample and augment the number of minority sample in difficult samples. Finally easy set and minority sample and majority samples in difficult set are combined to make up a new training set. Then I used standard scaler to standardize the data and digitize the sample label. Finally, the processed training set is used to train the classification model. K nearest neighbor , decision tree, logistic regression , and naive bayes are used as classifiers for classification models. Then the model is evaluated by a test set and identify most accurate model for detect intrusion.

(1) **Dataset**: The system use classic CSECIC-IDS2018 as benchmark datasets.

(2) **Preprocessing**: When the dataset is extracted, part of the data contains some noisy data, duplicate values, missing values,etc. due to extraction errors or input errors. Therefore,first perform the data preprocessing.

(3)**Training set**: Training set is processed for data balancing using DSSTE algorithm. The training set is divided into difficult set and easy set.

(4) **Difficult set**: The sample in the near neighbor set are very similor, making it very difficult for the classifier to learn the difference between the categories. So samples in the near neighbor set are difficult samples. Zoom in and zoom out the minority samples in difficult set. Compress the majority samples in difficult set.

(5) **Easy set** : The far neighbor set as easy set.

(6) **New Training set** : Easy set and majority and minority samples in the difficult set are combined to form a new training set.

(7) **Standard Scaler** : Standardize and digitize the sample labels.

(8) **Classification model** : Training set is used to train the classification model. Then the model is evaluated by the test set.

4.1 Dataset

CSE-CIC-IDS2018 on AWS A collaborative project between the Communications Security Establishment (CSE) the Canadian Institute for Cybersecurity (CIC).

In CSE-CIC-IDS2018 dataset, use the notion of profiles to generate datasets in a systematic manner, which will contain detailed descriptions of intrusions and abstract distribution models for applications, protocols, or lower level network entities. These profiles can be used by agents or human operators to generate events on the network. Due to the abstract nature of the generated profiles, then it can apply them to a diverse range of network protocols with different topologies.

4.2 Techniques used for feature selection

4.2.1 Feature selection with Random forest

Why we need feature selection techniques ?

• Reduced computational complexity.

• Increased performance of learning algorithm.

• Eliminates redundant information.

• Improves generalization and understanding of data.

RANDOM FOREST

Random Forest is an excellent supervised learning algorithm that can train a model to predict which classification results in a certain sample type belong to based on a given dataset's characteristic attributes and classification results. Random Forest is based on a decision tree and adopts the Bagging(Bootstrap aggregating) method to create different training sample sets. The random subspace division strategy selects the best attribute from some randomly selected attributes to split internal nodes. The various decision trees formed are used as weak classifiers, and multiple weak classifiers form a robust classifier, and the voting mechanism is used to classify the input samples. After a random forest has established a large number of decision trees according to a certain random rule when a new set of samples is input, each decision tree in the forest makes a prediction on this set of samples separately, and integrates the prediction results of each tree, get a final result.

Random forests are one the most popular machine learning algorithms. They are so successful because they provide in general a good predictive performance, low overfitting, and easy interpretability. This interpretability is given by the fact that it is straightforward to derive the importance of each variable on the tree decision. In other words, it is easy to compute how much each variable is contributing to the decision.Feature selection using Random forest comes under the category of Embedded methods. Embedded methods combine the qualities of filter and wrapper methods. They are implemented by algorithms that have their own built-in feature

selection methods. Some of the benefits of embedded methods are :

•They are highly accurate.

•They generalize better.

•They are interpretable.

4.3 Techniques used for classification

Classification is a form of supervised machine learning in which you train a model to use the features (the x values in our function) to predict a label (y) that calculates the probability of the observed case belonging to each of a number of possible classes and predicting an appropriate label. In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the input data and then uses this learning to classify new observations. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class.

•K Nearest Neighbors Classifier Model.

•Logistic Regression Model.

•Naive Baye Model.

• Decision Tree Model.

4.3.1 K Nearest Neighbors Classifier Model

The k-neighbors is commonly used and easy to apply classification method which implements the k neighbors queries to classify data. It is an instant-based and non-parametric learning method. In this method, the classifier learns from the instances in the training dataset and classifies new input by using the previously measured scores.

KNN classifies the new data points based on the similarity measure of the earlier stored data points. This algorithm finds the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (for classification) or averages the labels (for regression). While the KNN algorithm can be relatively easy to use and train, the accuracy of the KNN classifier will depend on the quality of the data and the specific K value chosen. When implementing a KNN classifier, data scientists must also decide how many neighbors to consider. In other words, we need to consider the optimal number of neighbors and how it impacts our classifier. The optimal K value (the number of neighbors considered) will impact the prediction model. Different data sets have different requirements.

KNN is best applied to datasets when they are labelled, noise-free, and relatively small. Given the classifications of data points in a training set, the algorithm can classify future unknown data based on this information. In addition, data sets with excess features that don't contribute to the classification of the data points may cause the algorithm to miss patterns in the data. These noise in the data set can

include extraneous data points that don't relate with the rest of the dataset and features that don't help in identifying the classification. Because the KNN algorithm is instance based, meaning no explicit training step is required, the training stage is relatively fast as compared to other methods. Therefore, with datasets with homogeneous features and few outliers and missing values, the KNN classifier can prove to be an accurate classifier.

4.3.2 Logistic Regression Model

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.
Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types

(1) **Binary or Binomial**

In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

(2) **Multinomial**

In such a kind of classification, dependent variable can have 3 or more possible unordered types or the types having no quantitative significance. For example,

these variables may represent “Type A” or “Type B” or “Type C”.

(3) **Ordinal**

In such a kind of classification, dependent variable can have 3 or more possible ordered types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”, “Excellent” and each category can have the scores like 0,1,2,3. Logistic Regression Assumptions are;
In case of binary logistic regression, the target variables must be binary always and the desired outcome is represented by the factor level 1.There should not be any multi-collinearity in the model, which means the independent variables must be independent of each other .We must include meaningful variables in our model. We should choose a large sample size for logistic regression.

4.3.3 Naive Baye Model

Naive Bayes is one such algorithm in classification that can never be overlooked upon due to its special characteristic of being “naive”. It makes the assumption that features of a measurement are independent of each other.
According to Bayes Theorem, the various features are mutually independent. For two independent events, P(A,B) = P(A)P(B). This assumption of Bayes Theorem is probably never encountered in practice, hence it accounts for the “naive” part in Naive Bayes. Bayes’ Theorem is stated as: P(a—b) = (P(b—a) * P(a)) / P(b). Where P(a—b) is the probability of a given b.

4.3.4 Decision Tree model

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.

4.4 DSSTE ALGORITHM

In imbalanced network traffic, different traffic data types have similar representations, especially minority attacks can hide among a large amount of normal traffic, making it difficult for the classifier to learn the differences between them during the training process. In the similar samples of the imbalanced training set, the majority class is redundant noise data. The number is much larger than the minority class, making the classifier unable to learn the distribution of the minority class and compress the majority class. The minority class discrete attributes remain constant, and there are differences in continuous attributes. Therefore, the minority class's continuous attributes are zoomed to produce data that conforms to the true distribution. Therefore, the proposed system use DSSTE algorithm to reduce the imbalance.

Algorithm 1 DSSTE Algorithm

Input: Imbalanced training set S, scaling factor K

Output: New training set SN

1: Step1: Distinguish easy set and difficult set

2: Take all samples from S and set it as SE

3: for each sample SE do

4: Compute its K nearest neighbors

5: Remove whose most K nearest neighbor samples are of different classes from SE

6: end for

7: Easy set SE, difficult set SD = S SE

8: Step2: Compress the majority samples in difficult set by the cluster centroid

9: Take all the majority samples from SD and set it as SMaj

10: Use KMeans algorithm with K cluster

11: Use the coordinates of the K cluster centroids replace the majority samples in SMaj

12: Compressed the majority samples set SMaj

13: Step3: Zoom augmentation

14: Take the minority samples from SD and set it as SMin

15: Take the Discrete attributes from SMin and set it as XD

16: Take the Continuous attributes from SMin and set it as XC

17: Take the Label attributes from SMin and set it as Y

18: for n range(K,K +numberSMin.shape[0]) do // zoom range is [1 1 K, 1 +1K], SMin.shape[0] is number of samples in SMin

19: XD1 = XD

20: XC1 = XC × (1 1n)

21: XD2 = XD

22: XC2 = XC × (1 +1n)

23: SZ append [concat(XD1, XC1, Y), concat(XD2, XC2, Y)]

24: end for

25: New training set SN = SE + SMaj + SMin + SZ

Functionality: First, the imbalanced training set to divide into near-neighbor set and far-neighbor set by Edited Nearest Neighbor(ENN) algorithm. The samples in the near-neighbor set are highly similar, making it very difficult for the classifier to learn the differences between the categories, so refer to the samples in the near-neighbor set as difficult samples and the far-neighbor set as easy samples. Next, step to zoom in and out the minority samples in difficult set. Finally, the easy set and minority in difficult set are combined with its augmentation samples to make up a new training set. The system use the K Nearest neighbors algorithm as the scaling factor of the entire algorithm. When scaling factor K increases, the number of difficult samples increases, and the compression rate of the majority of samples and the synthesis rate of the minority of class also increase.

4.5 DATA PREPROCESSING

When the dataset is extracted, part of the data contains some noisy data, duplicate values, missing values, infinity values, etc. due to extraction errors or input errors. Therefore, first perform data preprocessing. The main work is as follows.

(1) Duplicate values: delete the sample's duplicate value, only keep one valid data.

(2) Outliers: in the sample data, the sample size of missing values(Not a Number, NaN) and Infinite values(Inf) is small, so we delete this.

(3) Features delete and transform: In CSE-CIC-IDS2018, delete features such as "Timestamp", "Destination Address", "Source Address", "Source Port", etc.

4.6 EXPERIMENTAL PARAMETERS

The proposed method uses the Python and completes related experiments on the Google Colaboratory platform.

4.6.1 Google Colaboratory

Google Colaboratory also known as 'Google Colab' is a Free cloud based service hosted by Google which allows to write and execute Python in the browser and gives access to Graphical Processing Unit (GPU) and Tensor Processing Unit (TPU).Colab can be import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code.

4.6.2 Python

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a ”batteries included” language due to its comprehensive standard library.Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including metaprogramming and metaobject.Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It uses dynamic name resolution (late binding), which binds method and variable names during program execution.

Chapter 5

Experimental Results and Discussions

5.1 EVALUATION METRICS

Use the Accuracy, Prediction, Recall, and F1-Score to evaluate the experimental model's performance. These evaluation criteria reflect the performance of the intrusion detection system's flow recognition accuracy rate, and false alarm rate. The combination of the model prediction results and the true label is divided into four types: False Negative(FN), a positive sample, which is mistakenly judged as a negative sample. False Positive(FP), negative samples are misjudged as positive samples. True Negative(TN), actually negative samples, are correctly judged as negative samples. True Positive(TP), actually positive samples, are judged as the positive sample. These metrics are calculated according to Equations 4-7.

(5.0)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$F1\_Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$$

5.1.1 EXPERIMENTAL RESULTS

In this experiments, first explored the classifier's performance on the training set treated with different deflation factors. In the proposed DSSTE algorithm, there is a parameter scaling factor of K. When K increases within a certain range,the number of difficult samples will also increase, but when K exceeds the range, the number of difficult samples will constantly be constant. However, the majority compression and the minority augmentation in the difficult samples will increase with K change. The proposed system processed the training set in CSE-CICIDS2018 using different scaling factors K. i performed experiments on the proposed 4 classifiers, and performance was evaluated using the average F1-Score of each classifier

5.1.2 Performance Metrics

Performance analysis is done to identify the best model having the highest detection rate. The general evaluation metrics such as Accuracy, Precision, Recall, F1 score and confusion Matrix are used. High accuracy here indicates the enhanced detection rate and reduced false alarm rate.

The performance indicators includes: -

• True Positive (TP) is the number of correct classifications of attack category.

• True Negative (TN) is the number of correct classifications of normal category.

• False Positive (FP) is the number of incorrect classifications of attack category i.e., normal category wrongly classified as intrusive.

• False Negative (FN) is the number of incorrect classifications of normal category i.e., attack category wrongly classified as normal.

Derived performance metrics includes: -

• Accuracy gives us the percentage of total number of samples that are correctly classified. Accuracy = T P + T N/T P + T N + F P + F N (5.1)

• Precision The ratio of correctly predicted intrusive traffic to the total intrusive attacks.

Precision = T P /T P + F P (5.2) • Recall The performance of the model on detecting intrusive attacks.

Recall = T P /T P + F N (5.3)

• F1 Score : The accuracy of the model in whole dataset. It is the harmonic mean between precision and recall.

F1Score = 2 P recision Recall/P recision + Recall

• Confusion Matrix : The confusion matrix is utilized to determine the model's learning requirements. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the components of the confusion matrix. The performance results of the experimental studies conducted to identify the normal and attack categories obtained with intelligent IDS were tested using a confusion matrix. This matrix includes both predicted and actual data.

5.2 Naive Baye Classifier Model Result

```
Model Accuracy:
 0.906721354855782

Confusion matrix:
 [[2981  517]
 [ 188 3872]]

Classification report:
                precision    recall  f1-score   support

      anomaly       0.94      0.85      0.89      3498
       normal       0.88      0.95      0.92      4060

     accuracy                           0.91      7558
    macro avg       0.91      0.90      0.91      7558
weighted avg       0.91      0.91      0.91      7558
```

Figure 5.1: Naive Baye Classifier Model Result

5.3 LogisticRegression Model Result

```
Model Accuracy:
 0.9551468642498016

Confusion matrix:
 [[3297  201]
 [ 138 3922]]

Classification report:
                precision    recall  f1-score   support

     anomaly       0.96      0.94      0.95      3498
      normal       0.95      0.97      0.96      4060

    accuracy                           0.96      7558
   macro avg       0.96      0.95      0.95      7558
weighted avg       0.96      0.96      0.96      7558
```

Figure 5.2: Logistic Regression Classifier Model result

5.4 K Nearest Neighbors Classifier Model Result

```
Model Accuracy:
 0.9916644614977508

Confusion matrix:
 [[3458   40]
 [  23 4037]]

Classification report:
                precision    recall  f1-score   support

     anomaly       0.99      0.99      0.99      3498
      normal       0.99      0.99      0.99      4060

    accuracy                           0.99      7558
   macro avg       0.99      0.99      0.99      7558
weighted avg       0.99      0.99      0.99      7558
```

Figure 5.3: K Nearest Neighbors Classifier Model Result

5.5 Decision Tree Model Result

```
Model Accuracy:
 0.9947075946017465

Confusion matrix:
 [[3483   15]
 [  25 4035]]

Classification report:
                precision    recall  f1-score   support

     anomaly       0.99      1.00      0.99      3498
      normal       1.00      0.99      1.00      4060

    accuracy                           0.99      7558
   macro avg       0.99      0.99      0.99      7558
weighted avg       0.99      0.99      0.99      7558
```

Decision Tree Model Result

5.6 Accuracy Comparison Between Different Classification model

Classification Model	Accuracy
Naive Bayes	90%
Logistic Regression	95%
Kneighbors	99%
Decision Tree	99%

Table 5.1: Accuracy Comparison Between Different Classification model

From the above results, it is very clear that Decision Tree and K Nearest Neighbors Classification model algorithms in machine learning are better for detecting intrusion in the network.

Chapter 6

Conclusion

As network intrusion continues to evolve, the pressure on network intrusion detection is also increasing. In particular, the problems caused by imbalanced network traffic make it difficult for intrusion detection systems to predict the distribution of malicious attacks, making cyberspace security face a considerable threat.

This paper proposed a novel Difficult Set Sampling Technique(DSSTE) algorithm, which enables the classification model to strengthen imbalanced network data learning. A targeted increase in the number of minority samples that need to be learned can reduce the imbalance of network traffic and strengthen the minority's learning under challenging samples to improve the classification accuracy. The proposed system use three classical classification methods in machine learning.

Chapter 7

Future Scope

In the future directly use the deep learning model for feature extraction and model training on the original network traffic data, performance the advantages of deep learning in feature extraction, reduce the impact of imbalanced data and achieve more accurate classification.

REFERENCES

[1] S. Bhattacharya, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu,M. Alazab, and U. Tariq, "A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU," Electronics,vol. 9, no. 2, p. 219, Jan. 2020.

[2] P. Bedi, N. Gupta, and V. Jindal, "Siam-IDS: Handling class imbalance problem in intrusion detection systems using siamese neural network,"Procedia Comput. Sci., vol. 171, pp. 780–789, 2020.

[3] G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," Comput. Netw.,vol. 159, pp. 96–109, Aug. 2019.

[4] A. K. Verma, P. Kaushik, and G. Shrivastava, "A network intrusion detection approach using variant of convolution neural network," in Proc. Int.Conf. Commun. Electron. Syst. (ICCES), Jul. 2019, pp. 409–416.

[5] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system,"IEEE Access, vol. 7, pp. 94497–94507, 2019.

[6] A. Raghavan, F. D. Troia, and M. Stamp, "Hidden Markov models with random restarts versus boosting for malware detection,"J. Comput. Virol.Hacking Techn., vol. 15, no. 2, pp. 97–107, Jun. 2019.

[7] J.-T. Wang and C.-H. Wang, "High performance WGAN-GP based multiple-category network anomaly classification system," in Proc. Int.Conf. Cyber Secur. Emerg. Technol. (CSET), Oct. 2019, pp. 1–7.

[8] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," Cluster Comput.,vol. 22, pp. 949–961, 2019..

[9] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh,"Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," IEEE sensors Lett., vol. 3, no. 1,Jan. 2019, Art. no. 7101404.

[10] A. Ismail, S. A. Ahmad, A. C. Soh, K. Hassan, and H. H. Harith, "Improving convolutional neural network (CNN) architecture (miniVGGNet) with batch normalization and learning rate decay factor for image classification," Int. J. Integr. Eng., vol. 11, no. 4, pp. 1–9, 2019.

[11] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Trans. Emerg. Topics Comput. Intell.,vol. 2, no. 1, pp. 41–50, Feb. 2018.

[12] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends indeep learning based natural language processing [review article]," IEEE Comput. Intell. Mag., vol. 13, no. 3, pp. 55–75, Aug. 2018.

[13] X. Lei and Y. Xie, "Improved XGBoost model based on genetic algorithm for hypertension recipe recognition," Comput. Sci, vol. 45, pp. 476–481,2018.

[14] B. Yan and G. Han, "LA-GRU: Building combined intrusion detection model based on imbalanced learning and gated recurrent unit neural network," Secur. Commun. Netw., vol. 2018, pp. 1–13, Aug.2018.

[15] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy, 2018, pp. 108–116.

[16] H. Shapoorifard and P. Shamsinejad, "Intrusion detection using a novel hybrid method incorporating an improved KNN," Int. J. Comput. Appl.,vol. 173, no. 1, pp. 5–9, Sep. 2017.

[17] X. Ma and W. Shi, "AESMOTE: Adversarial reinforcement learning with SMOTE for anomaly detection," IEEE Trans. Netw. Sci. Eng., early access, Jun. 24, 2020, doi:

[18] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT," Sensors, vol. 17, no. 9, p. 1967, Aug. 2017.