Improving Network Intrusion Detection System using Imbalance Reduction Techniques

Submitted
in fulfillment of
the requirement for the Degree of Bachelor Of Technology

by

Shivani Pawar
(181071046)

Mounil Shah
(181070058)

Mona Gandhi
(181071021)

Ketaki Urankar
(181071069)

Under the Guidance of

Prof. Vaibhav D. Dhore

DEPARTMENT OF COMPUTER ENGINEERING AND INFORMATION TECHNOLOGY
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE
(An Autonomous Institute Affiliated to Mumbai University)
(Central Technological Institute, Maharashtra State)
Matunga, MUMBAI - 400019

A.Y. 2021-2022

DECLARATION OF STUDENT

I declare that the work embodied in Stage-II of this Project titled ***"Improving Network Intrusion Detection System using Imbalance Reduction Techniques"*** form my own contribution of work under the guidance of Prof. Vaibhav D. Dhore at the Department of Computer Engineering, Veermata Jijabai Technological Institute, Mumbai. The report reflects the work done during the period of Stage-II.

Shivani Pawar
Roll No.- 181071046
Date:
Place: VJTI, Mumbai

Mounil Shah
Roll No.- 181070058
Date:
Place: VJTI, Mumbai

Mona Gandhi
Roll No.- 181071021
Date:
Place: VJTI, Mumbai

Ketaki Urankar
Roll No.- 181071069
Date:
Place: VJTI, Mumbai

CERTIFICATE

This is to certify that,

Shivani Pawar	(181071046)
Mounil Shah	(181070058)
Mona Gandhi	(181071021)
Ketaki Urankar	(181071069)

students of B. Tech. (Computer Engineering), Veermata Jijabai Technological Institute, Mumbai have successfully completed the Stage-II of the project titled ***"Improving Network Intrusion Detection System using Imbalance Reduction Techniques"*** under the guidance of Prof. Vaibhav D. Dhore.

Prof. Vaibhav D. Dhore

Supervisor

Dr. Mahesh R. Shirole

Head of Computer Engineering Department

Place: *Veermata Jijabai Technological Institute, Mumbai.*
Date: 25/05/2022

Abstract

Due to the massive expansion in the usage of the Internet in everyday life, intruders have been able to attempt to breach the security principles of availability, confidentiality, and integrity. As a result, work has been done to develop techniques such as the Network Intrusion Detection System (NIDS), which monitors and analyzes network flow and attacks detection. However, malicious cyber-attacks can often lurk in enormous amounts of benign data in unbalanced traffic, making it hard for the NIDS to ensure the accuracy and timeliness of detection. In this study, existing strategies for reducing imbalances in a methodical manner are explored and compared. The issues and potential for reducing the imbalance, resulting in the development of a novel imbalance reduction technique that improves on state-of-the-art NIDS approaches have been invesetigated. Comparison between the existing imbalance reduction techniques, including undersampling techniques such as ENN, Tomek links, Cluster Centroids, IHT, and Random undersampling, oversampling techniques such as SMOTE, borderlineSMOTE, Adasyn, and random oversampling, and ensemble techniques such as the recently introduced DSSTE algorithm has been made. An ensemble of all undersampling strategies, an ensemble of all oversampling techniques, and an ensemble of selected oversampling methods are also compared. These strategies are tested on the classic intrusion dataset NSL-KDD as well as the more recent and comprehensive intrusion dataset CSE-CIC-IDS2018. For this, the traditional machine learning models such as Random Forest (RF), Decision Tree (DT), and XGBoost have been used. The results obtained show that certain oversampling and undersampling techniques are more suitable for particular datasets, irrespective of the models trained. Other than the pre-existing techniques, the ensemble techniques proposed in this report have shown better, if not comparable results.

Contents

Chapter 1

Introduction

With the expansion of the internet, the usage has increased massively in the previous few years. Accessibility to internet is now almost a fundamental need. With this explosion, there has been a rise in the security breaches and cyber attacks. These attacks include malicious intrusion into host network, stealing of valuable data, identity theft, etc. This has led to a need to improve the current defense systems. New intrusion detection methods are being developed in order to detect and if there need be, even prevent malicious access to users' network.

A hardware or software programme that monitors a network or systems for malicious activity or policy breaches is known as an intrusion detection system (IDS; sometimes known as an intrusion prevention system or IPS).

IDS types include a wide range of applications, from single computers to big networks. Network intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS) are the two most popular types. An HIDS is a system that monitors essential operating system files, whereas an NIDS is a system that analyses incoming network traffic. IDS can also be classified using a detection strategy.

Signature-based detection (recognising harmful patterns, such as malware) and anomaly-based detection are the most well-known variations. Another popular form is detection based on a person's reputation (recognizing the potential threat according to the reputation scores). Some IDS products offer the capacity to respond to intrusions that have been detected. Intrusion prevention systems are often referred to as systems having reaction capabilities. Intrusion detection systems can also be enhanced with specialised tools to serve specific purposes, such as attracting and characterising malicious traffic using a honeypot.

Classification of Network Intrusion Detection Systems (NIDS)

Network-based intrusion prevention system (NIPS): monitors the entire network for suspicious traffic by analyzing protocol activity.

Wireless intrusion prevention system (WIPS): monitor a wireless network for suspicious traffic by analyzing wireless networking protocols.

Network behavior analysis (NBA): examines network traffic to identify threats that generate unusual traffic flows, such as distributed denial of service (DDoS) attacks, certain forms of malware and policy violations.

Host-based intrusion prevention system (HIPS): an installed software package which monitors a single host for suspicious activity by analyzing events occurring within that host.

1.1 Project Idea

The project aims to reduce the imbalance problem in the CSE CIC IDS 2018 dataset, the latest dataset use for NIDS, using various ensemble of imbalance reduction techniques. The project also intends to improve upon the ability to identify various classes of attacks with better efficiency. A pipeline is designed wherein initially various individual and ensembles of imbalance reduction methods are applied to the dataset and subsequently the data is trained on RandomForest model, SVM model and XGBoost model. The results of applying several ensembles are analyzed and compared, and the best technique of imbalance methods is consequently deduced from the results.

1.2 Motivation of the Project

With the rise in popularity of the internet, there has also been a rise in the cyber attacks carried across it. To tackle these cyber attacks, Network Intrusion Detection Systems (NIDS) are used, which must accurately identify the attacks and protect the applications.

CSE-CIC-IDS 2018 is the latest dataset for evaluating Intrusion Detection Systems (IDS). However, the dataset has multiclass imbalance. In such imbalanced network traffic, malicious cyber attacks can often hide in large amounts of normal data. The algorithms are rendered inefficient at distinguishing the classes in a highly imbalanced environment.

1.3 Technical Keywords

Dataset: NSL-KDD; CSE-CIC-IDS2018.
Technical: Machine learning, Deep Learning, Random Forest, Support Vector Machine, XGBoost, SMOTE, Random undersampling, Random oversampling, Cluster Centroid, Edited Nearest Neighbour, Tomek Links.

1.4 Problem Statement

Comparing various imbalance reduction methods – undersampling and oversampling – on NIDS datasets and improving by using their respective ensembles. Using the improved imbalance reduction techniques to get better results on multiclass classification on the CSE-CIC-IDS 2018 dataset. Comparing the results thus obtained with DSSTE and other imbalance reduction techniques.

1.5 Objectives

- To propose an efficient method for reducing imbalance in dataset.
- To obtain results for the pre-existing imbalance reduction techniques and comparing the techniques for various machine learning models.
- To improve upon the ability to correctly classify the various classes of CSE-CIC-IDS 2018 dataset.

1.6 Plan of Project Execution

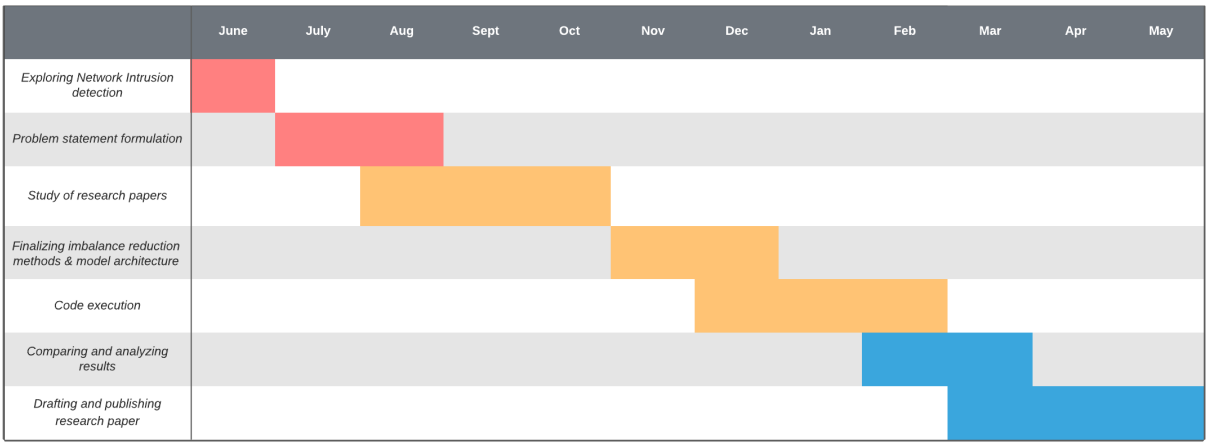

Figure 1.1: *Project Timeline*

Chapter 2

Literature Survey

2.1 KDD99 Dataset

The KDD99 was created by MIT and utilized in the International Knowledge Discovery and Data Mining Tool Competition [20]. The benchmark dataset for Intrusion Detection System (IDS) was KDD99 released by DARPA [18]. The dataset was prepared in 1999 and has become the most widely used dataset for the evaluation of anomaly detection although KDD99 dataset is more than 20 years old [21]. KDD99 dataset consists of 4,898,431 instances each of which consists of 42 features. Table 1 shows KDD99 dataset features.

KDD99 contents a total of 22 training attacks types and one normal, with 17 additional types in the testing data only. The 41 features labelled as either special attack type (DOS, U2R, R2L, and Probe) or normal. It is believed that attacks can be detected with the knowledge earned from the registered

2.2 UNSW-NB 15 dataset

The UNSW-NB 15 dataset was developed in the Cyber Range Lab of UNSW Canberra using the IXIA PerfectStorm programme to generate a combination of real modern normal activities and synthetic contemporary attack behaviours. 100 GB of raw traffic was captured using the tcpdump programme (e.g., Pcap files). Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms are among the nine types of attacks in this dataset. To produce a total of 49 characteristics with the class label, the Argus and Bro-IDS tools are utilised, and twelve methods are built. The UNSW-NB15 features.csv file describes these features.

UNSW-NB15 1.csv, UNSW-NB15 2.csv, UNSW-NB15 3.csv, and UNSW-NB15 4.csv store a

total of two million and 540,044 records in four CSV files: UNSW-NB15 1.csv, UNSW-NB15 2.csv, UNSW-NB15 3.csv, and UNSW-NB15 4.csv. UNSW-NB15 GT.csv is the ground truth table, while UNSW-NB15 LIST EVENTS.csv is the list of events file. The training set has 175,341 records, while the testing set contains 82,332 records of various sorts, including attack and normal.

2.3 CIDDS-001-2017

The CIDDS-001 (Coburg Intrusion Detection DataSet) is a labelled flow-based dataset. This dataset developed for the evaluation purpose of Anomaly-based Network Intrusion Detection System (NIDS). CIDDS-001 dataset consists of unidirectional NetFlow data, it consists of traffic data from OpenStack environment having internal servers (backup, mail, file, and web) and External Servers External Server (file synchronization and web server), which is deployed on the internet to capture real-time and up-to-date traffic from the internet . CIDDS- 001 dataset consists of realistic normal and attacks traffic that allow for an important measurement of NIDS on Cloud environment. It is divided into four parts each is created during a week. It contains 14 features, the first 10 features are the default NetFlow features and the last four features are additional features. The CIDDS- 001 dataset contains 16 million flows. It was captured over a period of two weeks . Attack flows are captured in the dataset within four attacks types (suspicious, attacker, unknown, and victim) .

A lot of studies are being done on the development of effective NIDS using CIDDS-001 dataset. Here are some studies that used CIDDS-001 dataset: Rashid et al. (Rashid A, Siddique MJ, Ahmed SM. Machine and deep learning based comparative analysis using hybrid approaches for intrusion detection system. in 2020 3rd International Conference on Advancements in Computational Sciences (ICACS). IEEE; 2020.) introduced a comparative analysis on benchmark datasets NSL-KDD and CIDDS-001 using machine and deep learning algorithms. For getting optimal results, the hybrid feature selection and ranking methods was used. Six classification algorithms used such as SVM, Naïve Bayes, k-NN, Neural Networks, DNN, and DAE. The experimental results showed that k-NN, SVM, NN, and DNN classifiers achieved high performance on the NSLKDD dataset whereas k-NN and Naïve Bayes classifiers achieved high performance on the CIDDS-001 dataset. feature selection on KDD99, UNSW-NB15, and CIDDS-001 datasets. Mean Decrease Impurity (MDI), Random Forest Classifier (RFC), Stability Selection (SS), Recursive Feature Elimination (RFE), and Chi-square were used to get the score of each feature. Then, a simple voting method used to integrate feature selection methods. Decision Tree (DT), k-NN (k-nearest neighbor), SVM, and Multi-Layer Perception (MLP) are used for classification. The feature subsets with classification accuracy before and after the ensemble were compared. The experiment showed that the EFS achieved high accuracy in classification. Verma et al. (Singh Panwar S, Raiwani Y, Panwar LS. Evaluation of network intrusion detection with features selection and machine learning algorithms on CICIDS-2017 dataset. in International Conference on Advances in Engineering Science Management Technology (ICAESMT)- 2019, Uttaranchal University, Dehradun, India; 2019)discussed the statistical analysis and evaluation using the

CIDDS-001 dataset. Two techniques, K-NN and k-means clustering were used. On the basis of evaluation results, it concluded that both K-NN and k-means clustering perform well over CIDDS-001 dataset

2.4 NSL-KDD Dataset

As proposed in [1] new approach consists of merging feature selection and classification for multiple class NSL-KDD cup 99 intrusion detection dataset employing support vector machine (SVM). The objective is to improve the competence of intrusion classification with a significantly reduced set of input features from the training data. In supervised learning, feature selection is the process of selecting the important input training features and removing the irrelevant input training features, with the objective of obtaining a feature subset that produces higher classification accuracy. SVM classifier was applied on several input feature subsets of training dataset of NSL-KDD cup 99 dataset. The experimental results obtained showed the proposed method successfully bring 91 percent classification accuracy using only three features and 99 percent classification accuracy using 36 features, while all 41 training features achieved 99 percent classification accuracy.

In [2] the NSL-KDD data set is analysed and used to study the effectiveness of the various classification algorithms in detecting the anomalies in the network traffic patterns. Also, the relationship of the protocols available in the commonly used network protocol stack with the attacks used by intruders to generate anomalous network traffic, was analyzed. The analysis is done using classification algorithms available in the data mining tool WEKA. The study has exposed many facts about the bonding between the protocols and network attacks.

Since datasets used in intrusion detection are imbalanced, the accuracy of detecting two attack classes, R2L and U2R, is lower than that of the normal and other attack classes. In order to overcome this issue, [3] employs a hybrid approach. This hybrid approach is a combination of synthetic minority oversampling technique (SMOTE) and cluster center and nearest neighbor (CANN). Important features are selected using leave one out method (LOO). Results indicate that the proposed method improves the accuracy of detecting U2R and R2L attacks in comparison to the baseline paper by 94 percent and 50 percent, respectively.

Here [4] the analysis of KDD data set with respect to four classes which are Basic, Content, Traffic and Host in which all data attributes can be categorized has been presented. The analysis is done with respect to two prominent evaluation metrics, Detection Rate (DR) and False Alarm Rate (FAR) for an Intrusion Detection System (IDS). As a result of this empirical analysis on the data set, the contribution of each of four classes of attributes on DR and FAR is shown which can help enhance the suitability of data set to achieve maximum DR with minimum FAR

2.5 CSE-CIC-IDS 2018 dataset

In [5], classification performance in detecting web attacks in the recent CSE-CIC-IDS 2018 dataset is explored This study considers a total of eight random undersampling (RUS) ratios: no sampling, 999:1, 99:1, 95:5, 9:1, 3:1, 65:35, and 1:1. Additionally, seven different classifiers are employed: Decision Tree (DT), Random Forest (RF), CatBoost (CB), LightGBM (LGB), XGBoost (XGB), Naive Bayes (NB), and Logistic Regression (LR). For classification performance metrics, Area Under the Receiver Operating Characteristic Curve (AUC) and Area Under the Precision-Recall Curve (AUPRC) are both utilized.

In [6] CSE-CIC-IDS 2018 dataset to investigate ensemble feature selection on the performance of seven classifiers : Decision Tree (DT), Random Forest (RF), Naive Bayes (NB), Logistic Regression (LR), Catboost, LightGBM, or XGBoost. Its impact on AUC and F1 score is investigated.

A Hybrid Feature Selection approach that aims to reduce the prediction latency without affecting attack prediction performance by lowering the model's complexity has been proposed in [7].The proposed feature selection reduces the prediction latency ranging from 44.52% to 2.25% and the model building time ranging from 52.68% to 17.94% in various algorithms on the CIC-IDS 2018 dataset.

Two experiments, starting with the analysis of two modern intrusion detection datasets CIC-IDS2017 and CSE- CIC-IDS2018 by means of supervised machine learning have been detailed in [8]. An investigation into the generalizing capabilities of these supervised learners is conducted by exposing pre-trained models to unseen at- tack data from the other dataset.

[9] proposes an effective deep learning method, namely AE-IDS (Auto-Encoder Intrusion Detection System) based on random forest algorithm. This method constructs the training set with feature selection and feature grouping. After training, the model can predict the results with auto-encoder, which greatly reduces the detection time and effectively improves the prediction accuracy.

In [10], first, four unsupervised machine learning methods on two recent datasets are evaluated and then their generalization strength is defined using a novel inter-dataset evaluation strategy estimating their adaptability. Results show that all models can present high classification scores on an individual dataset but fail to directly transfer those to a second unseen but related dataset. Specifically, the accuracy dropped on average 25.63 percent in an inter-dataset setting compared to the conventional evaluation approach.

A DL-based intrusion model has been proposed in [10] especially focusing on denial of service (DoS) attacks. For the intrusion dataset, KDD CUP 1999 dataset (KDD), the most widely used dataset for the evaluation of intrusion detection systems (IDS), is used. In addition to KDD, CSE-CIC-IDS2018 which is the most up-to-date IDS dataset, is used. Model based on a Convolutional Neural Network (CNN) and evaluate its performance through comparison with an Recurrent Neural Network (RNN), is used. Furthermore, it suggests the optimal

CNN design for the better performance through numerous experiments.

The paper [11] provides an overview of CC and MCC paradigms and service models, also reviewing security threats in these contexts. Previous literature is critically surveyed, highlighting the advantages and limitations of previous work. Then, a taxonomy for IDS and classify CI-based techniques into single and hybrid methods, is defined.

A survey of deep learning approaches for cyber security intrusion detection has been presented in [12], the datasets used, and a comparative study. Specifically, a review of intrusion detection systems based on deep learning approachesis provided. The dataset plays an important role in intrusion detection, there- fore it describes 35 well-known cyber datasets and provide a classification of these datasets into seven categories; namely, network traffic-based dataset, electrical network-based dataset, internet traffic-based dataset, virtual private network-based dataset, android apps-based dataset, IoT traffic-based dataset, and internet-connected devices-based dataset. It analyzes seven deep learning models including recurrent neural networks, deep neural networks, restricted Boltzmann machines, deep belief networks, convolu- tional neural networks, deep Boltzmann machines, and deep autoencoders.

In [13], Long Short Term Memory (LSTM) to build a deep neural net-work model and add an Attention Mechanism (AM) to enhance the performance of the model, is used. The SMOTE algorithm and an improved loss function are used to handle the class-imbalance problem in the CSE-CIC-IDS2018 dataset. The ex-perimental results show that the classification accuracy of the model reaches 96.2 percent, which is higher than other machine learning algorithms. In addition, the class-imbalance problem is alleviated to a certain extent, making the proposed method have great practicality.

Six machine-learning-based IDSs by using K Nearest Neighbor, Random Forest, Gradient Boosting, Adaboost, Decision Tree, and Linear Discriminant Analysis algorithms have been proposed in [14]. To implement a more realistic IDS, an up-to-date security dataset, CSE-CIC-IDS2018, is used instead of older and mostly worked datasets. The selected dataset is also imbalanced. Therefore, to increase the efciency of the system depending on attack types and to decrease missed intrusions and false alarms, the imbalance ratio is reduced by using a synthetic data generation model called Synthetic Minority Oversampling TEchnique (SMOTE).

2.6 Imbalance Reduction

The effect of class imbalance on the benchmark NSL-KDD dataset is evaluated in [15] using four popular classification techniques and the results are analyzed.

Imbalance reduction techniques like rulelearning, using RIPPER, on highly imbalanced intrusion datasets have been focused upon in [16] with an objective to improve the true positive rate (intrusions) without significantly increasing the false positives.

The proposed method in [17] combines Synthetic Minority Over-sampling Technique (SMOTE) and Complementary Neural Network (CMTNN) to handle the problem of classifying imbalanced data. In order to demonstrate that the proposed technique can assist classification of imbalanced data, several classification algorithms have been used, which are Artificial Neural Network (ANN), kNearest Neighbor (k-NN) and Support Vector Machine (SVM).

A new combined IDM called LA-GRU based on a novel imbalanced learning method and gated recurrent unit (GRU) neural network is proposed in [18]. In the proposed model, a modified local adaptive synthetic minority oversampling technique (LA-SMOTE) algorithm is provided to handle imbalanced traffic, and then the GRU neural network based on deep learning theory is used to implement the anomaly detection of traffic.

It [19] proposes a novel Difficult Set Sampling Technique(DSSTE) algorithm to tackle th 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. To verify the proposed method, experiments on the classic intrusion dataset NSL-KDD and the newer and comprehensive intrusion dataset CSE-CIC-IDS2018 have been conducted.The classical classification models used: random forest(RF), Support Vector Machine(SVM), XGBoost, Long and Short-term Memory(LSTM), AlexNet, Mini-VGGNet. It compares the other 24 methods; the experimental results demonstrate that the proposed DSSTE algorithm outperforms the other methods.

In [20] Editing of the preclassified samples using the three-nearest neighbor rule followed by classification using the single-nearest neighbor rule with the remaining preclassified samples appears to produce a decision procedure whose risk approaches the Bayes' risk quite closely in many problems with only a few preclassified samples. The asymptotic risk of the nearest neighbor rules and the nearest neighbor rules using edited preclassified samples is calculated for several problems

It has been shown in [21] that a combination of over-sampling the minority (abnormal) class and under-sampling the majority (normal) class can achieve better classifier performance (in ROC space) than only under-sampling the majority class. This paper also shows that a combination of the method of over-sampling the minority class and under-sampling the majority class can achieve better classifier performance (in ROC space) than varying the loss ratios in Ripper or class priors in Naive Bayes

A novel adaptive synthetic (ADASYN) sampling approach has been presented in [22] for learning from imbalanced data sets. The essential idea of ADASYN is to use a weighted

distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn.

2.7 Imbalance Reduction Methods

2.7.1 Oversampling Imbalance Reduction Techniques

1. **Random Oversampling**
   Random oversampling involves randomly duplicating examples from the minority class and adding those examples to the training dataset.

   Examples from the training dataset are selected randomly with replacement. This means that examples from the minority class can be chosen and added to the new "more balanced" training dataset multiple times; which are selected from the original training dataset, added to the new training dataset, and then returned or "replaced" in the original dataset, allowing the examples to be selected again.

   This technique can be effective for those machine learning algorithms that are affected by a skewed distribution and where multiple duplicate examples for a given class can influence the fit of the model. This might include algorithms that iteratively learn coefficients, like artificial neural networks that use stochastic gradient descent. It can also affect models that seek good splits of the data, such as support vector machines and decision trees.

   It might be useful to tune the target class distribution. In some cases, seeking a balanced distribution for a severely imbalanced dataset can cause affected algorithms to overfit the minority class, leading to increased generalization error. The effect can be better performance on the training dataset, but worse performance on the holdout or test dataset.

2. **SMOTE**
   SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem. The method was proposed in a 2002 paper in the Journal of Artificial Intelligence Research. SMOTE is an improved method of dealing with imbalanced data in classification problems.

   It aims to balance class distribution by randomly increasing minority class examples using replication.

   SMOTE synthesises new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class. After the oversampling process, the data is reconstructed and several classification models can be applied for the processed data.

3. **Borderline SMOTE**
   A popular extension to SMOTE involves selecting those instances of the minority class that are misclassified, such as with a k-nearest neighbor classification model.

   Then, just those difficult instances,can be oversampled providing more resolution only where it may be required.

   These examples that are misclassified are likely ambiguous and in a region of the edge or border of decision boundary where class membership may overlap. As such, this modified to SMOTE is called Borderline-SMOTE and was proposed by Hui Han, et al. in their 2005 paper titled "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning."

   The authors also describe a version of the method that also oversampled the majority class for those examples that cause a misclassification of borderline instances in the minority class. This is referred to as Borderline-SMOTE1, whereas the oversampling of just the borderline cases in minority class is referred to as Borderline-SMOTE2.

4. **ADASYN**
   ADASYN is similar to SMOTE, and derived from it, featuring just one important difference. it will bias the sample space (that is, the likelihood that any particular point will be chosen for duping) towards points which are located not in homogenous neighborhoods.

   ADASYN uses the normal SMOTE algorithm on point not in homogenous neighborhoods. The result is a kind of hybrid between regular SMOTE and borderline1 SMOTE. This technique inherits the primary weakness of SMOTE, e.g. its ability to create innerpoint-outerpoint bridges. Whether or not the heavy focus on the outlier points is a good thing or not is application dependent, but overall ADASYN feels like a very heavy transformation algorithm, and e.g. one requiring that the underlying point cluster be sufficiently large, as imblearn doesn't provide any modifications to this algorithm for modulating its tendency to create point (as it does for SMOTE).

2.7.2 Undersampling Imbalance Reduction Techniques

1. **Random Undersampling**
   Random undersampling involves randomly selecting examples from the majority class to delete from the training dataset.

   This has the effect of reducing the number of examples in the majority class in the transformed version of the training dataset. This process can be repeated until the desired class distribution is achieved, such as an equal number of examples for each class.

   This approach may be more suitable for those datasets where there is a class imbalance although a sufficient number of examples in the minority class, such a useful model can be fit.

A limitation of undersampling is that examples from the majority class are deleted that may be useful, important, or perhaps critical to fitting a robust decision boundary. Given that examples are deleted randomly, there is no way to detect or preserve "good" or more information-rich examples from the majority class.

2. **Tomek Links**
   Tomek Links is one of a modification from Condensed Nearest Neighbors (CNN, not to be confused with Convolutional Neural Network) undersampling technique that is developed by Tomek (1976). Unlike the CNN method that are only randomly select the samples with its k nearest neighbors from the majority class that wants to be removed, the Tomek Links method uses the rule to selects the pair of observation (say, a and b) that are fulfilled these properties:

   The observation a's nearest neighbor is b. The observation b's nearest neighbor is a. Observation a and b belong to a different class. That is, a and b belong to the minority and majority class (or vice versa), respectively.

   This method can be used to find desired samples of data from the majority class that is having the lowest Euclidean distance with the minority class data (i.e. the data from the majority class that is closest with the minority class data, thus make it ambiguous to distinct), and then remove it.

3. **Cluster Centroid**
   To minimize the degree of imbalance, Data Mining and Feature Space Geometry has to be incorporated into the Classical Methodology of solving Machine Learning Classification Problems. There are many Data Mining approaches for Data Balancing. One such important approach is Cluster Centroid based Majority Under-sampling Technique (CCMUT).

   Cluster centroids is a method that replaces cluster of samples by the cluster centroid of a K-means algorithm, where the number of clusters is set by the level of undersampling

   In Majority Under-sampling, unimportant (or not-so-important) instances are removed among majority samples. In CCMUT, the demarcation of instances as important and unimportant is done by using the concept of Clustering on Feature-Space Geometry.

   Clustering is an Unsupervised Learning Approach. But CCMUT, only uses the concept of finding cluster centroid (clusters are created encircling data-points belonging to the majority class), as already instances are labelled. The cluster centroid is found by obtaining the average feature vectors for all the features, over the data points belonging to the majority class in feature space.

   After finding the cluster centroid of the majority class, the instance belonging to the cluster (majority class), which is farthest from the cluster centroid in feature space, is considered to be the most unimportant instance. On the contrary, the instance belonging to the majority class, that is nearest to the cluster centroid in feature space, is considered to be the most important instance.

   So, in CCMUT, instances belonging to the majority class are removed on the basis of their importance and number of samples to be under-sampled depends upon the

percentage of Under-sampling or CCMUT.

4. **Edited Nearest Neighbour**
   ENN method works by finding the K-nearest neighbor of each observation first, then check whether the majority class from the observation's k-nearest neighbor is the same as the observation's class or not. If the majority class of the observation's K-nearest neighbor and the observation's class is different, then the observation and its K-nearest neighbor are deleted from the dataset. In default, the number of nearest-neighbor used in ENN is $K = 3$.

   This method is more powerful than Tomek Links, where ENN removes the observation and its K-nearest neighbor when the class of the observation and the majority class from the observation's K-nearest neighbor are different, instead of just removing observation and its 1-nearest neighbor that are having different classes. Thus, ENN can be expected to give more in-depth data cleaning than Tomek Links.

5. **Instance Hardness Threshold**
   Instance hardness is the probability of an observation being miss classified. In other words, it is 1 - probability of the class.

   The idea is that the probabilities given by the estimator are related to the certainty for a sample to belong to the class. Therefore, a percentile of 0.0 would mean that all samples are selected while a percentile of 1.0 mean that a single sample is selected(the one with the maximum probability). So the threshold corresponds to select the N most certain samples to belong to class C as seen per the estimator. N is defined by the sampling-strategy parameter (e.g., the expected balancing ratio).

2.8 Literature Gap

As it can be observed from the preceding section, various imbalance reduction methods and techniques have been performed on several NIDS datasets, like KDD and NSL-KDD datasets. In comparison to these, much less research and experimentation has been performed on the CSE-CIC-IDS dataset. It takes a considerable amount of time to apply various imbalance techniques and to train machine learning models on the CSE-CIC-IDS 2018 dataset due to its enormous size and quantity, and, hence, better methods or frameworks need to be devised to apply these methods optimally.

The CSE-CIC-IDS 2018 dataset is an improvement on the CIC-IDS 2017 dataset. A lot of research and study has gone into training models on the CIC-IDS 2017 dataset. However, there is still scope for the study of CSE-CIC-IDS 2018 dataset.

With respect to imbalance reduction techniques, various methods have been used on the NSL-KDD dataset. Same is not the case with the CSE-CIC-IDS 2018 dataset. Only a few methods have been used while training the respective machine learning and/or deep learning

models. Various undersampling and oversampling methods can be explored to get better results upon training models using the resultant datasets.

Apart from the pre-existing imbalance reduction methods, another idea for reducing class imbalance involves using various methods at disposal and forming their ensemble. This method of reduction can helps get the best out of all the techniques used.

Chapter 3

Proposed System

3.1 System Architecture Diagram

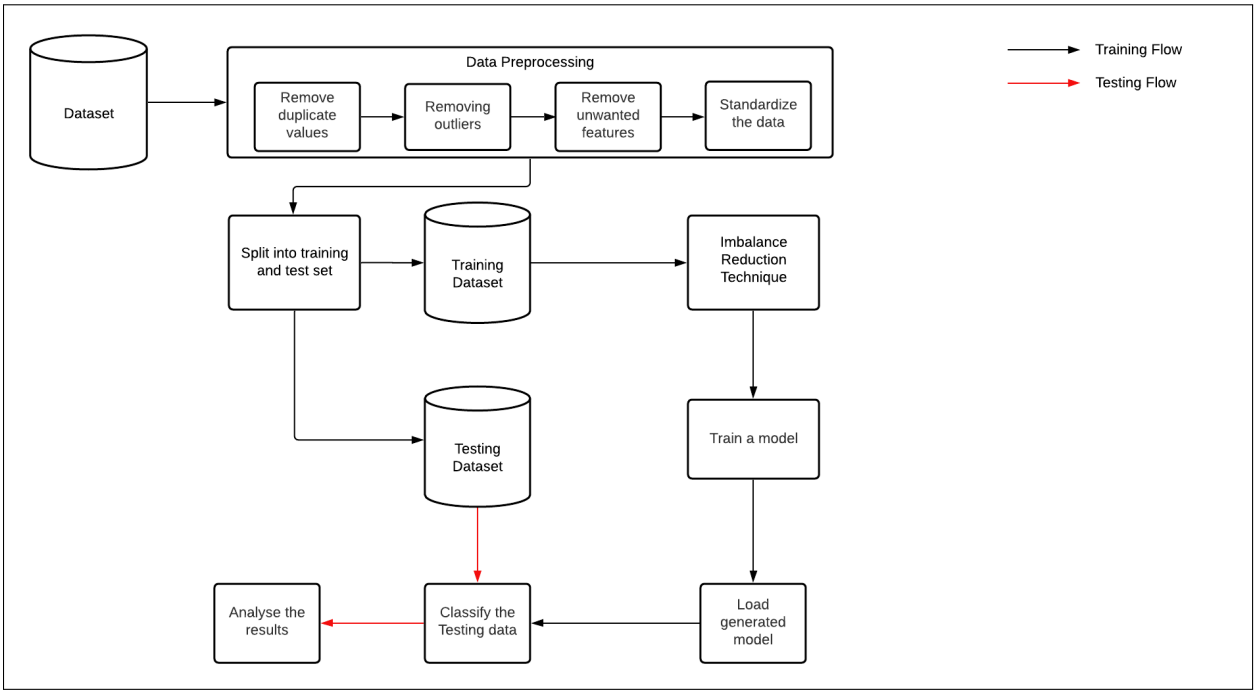

Figure 3.1: *System Architecture*

3.2 Methodologies of Problem Solving

3.2.1 Dataset Description

CSE-CIC-IDS 2018 dataset

CSE-CIC-IDS2018 is an intrusion detection dataset created by the Canadian Institute of Cyber Security (CIC) on AWS (Amazon Web Services) in 2018. It is also the latest and comprehensive intrusion dataset currently publicly available. CSE-CIC-IDS2018 is a dataset collected for launching real attacks. It is an improvement based on the CSE-CIC-IDS2017 dataset. In addition to the basic criteria, it offers the following advantages:

- The number of duplicate data is very low,
- Uncertain data is nearly absent
- The dataset is in a CSV format, so it is ready to use without processing

The dataset contains different types of attack scenarios : DOS, BruteForce, Infiltration, Bot and Web attacks. The data was gathered using an attacking infrastructure having 50 machines and the victim organization having 5 departments, and 420 machines and 30 servers. The network traffic and system logs of each machine, along with 80 features is extracted from the captured traffic. The dataset is enormous with 83 features, 1,62,32,943 records and 13 attack types with 1 benign type.The dataset suffers from the problem of imbalance.

It contains the necessary standards for the attack dataset and covers various known attack types. The dataset contains six different attack scenarios: Brute Force, Botnet, DoS, DDoS, Web Attacks, and Infiltration. Each sample in CSE-CIC-IDS2018 includes 83 features.

NSL-KDD dataset

The NSL-KDD dataset was created in 2009 to solve problems related to irregular data in the KDD Cup 99 dataset. The reliability of the systems developed in the previous years was questioned, as there were no accurate datasets for IDSs. The NSL-KDD dataset has important advantages over the original KDD Cup99 dataset:

- Unnecessary records in training data have been eliminated; it contains important records in the KDD Cup99 dataset
- It doesn't have duplicate data

Traffic Class	Record Count
Benign	13484708
HOIC	686012
LOIC-HTTP	576191
Hulk	461191
Bot	286191
FTP-BruteForce	193360
SSH-BruteForce	187589
Infilteration	161934
SlowHTTPTest	139890
GoldenEye	41508
Slowloris	10990
LOIC-UDP	1730
BruteForce-Web	611
BruteForce-XSS	230
SQL-Injection	11
Total	16232943

- More homogeneous distribution
- The number of records in the training and test sets is proportionally distributed,

The NSL-KDD dataset contains a feature map with 42 features, which are grouped under four categories:

1. General features
2. Content features
3. Server-based traffic features
4. Time-dependent traffic features

Attacks in the NSL-KDD dataset are divided into four different categories: DoS, Probe, U2L, and R2L. In addition to these attacks, there is a single Normal/Benign category.

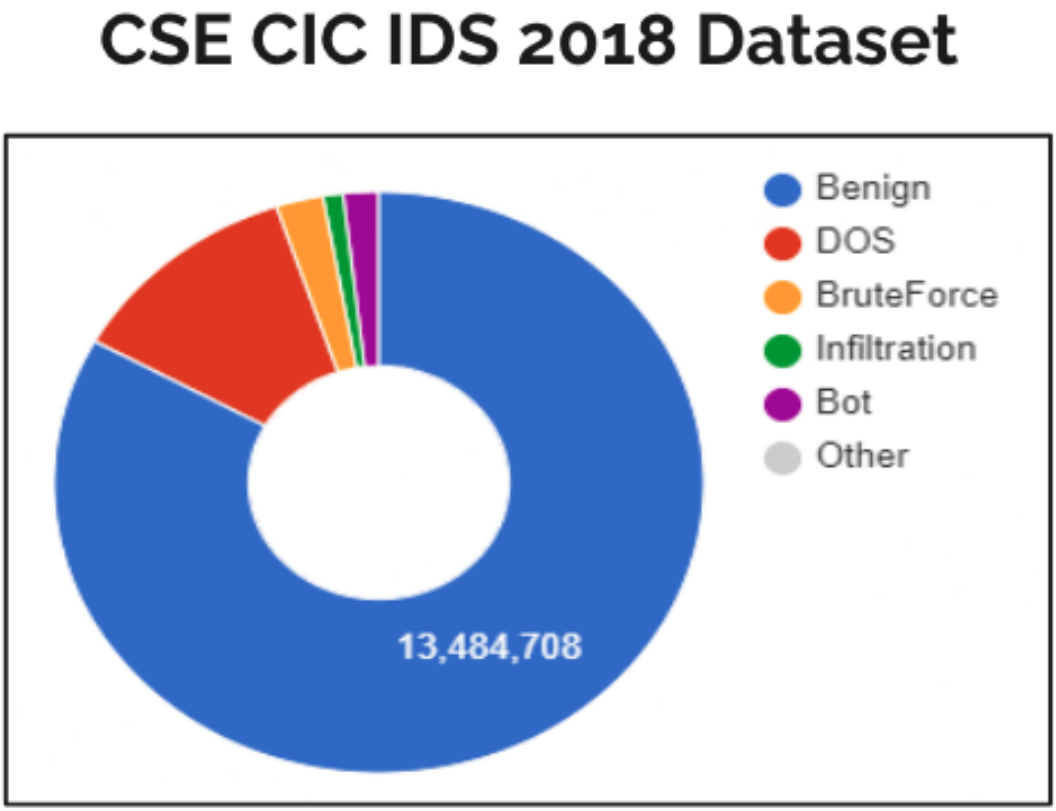

Figure 3.2: *Classwise Distribution in CSE-CIC-IDS2018*

Traffic Class	Record Count
Normal	13449
Probe	2289
DOS	9234
U2R	11
R2L	209
Total	25192

3.2.2 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, data preprocessing is performed. These are the four simple steps for preprocessing:

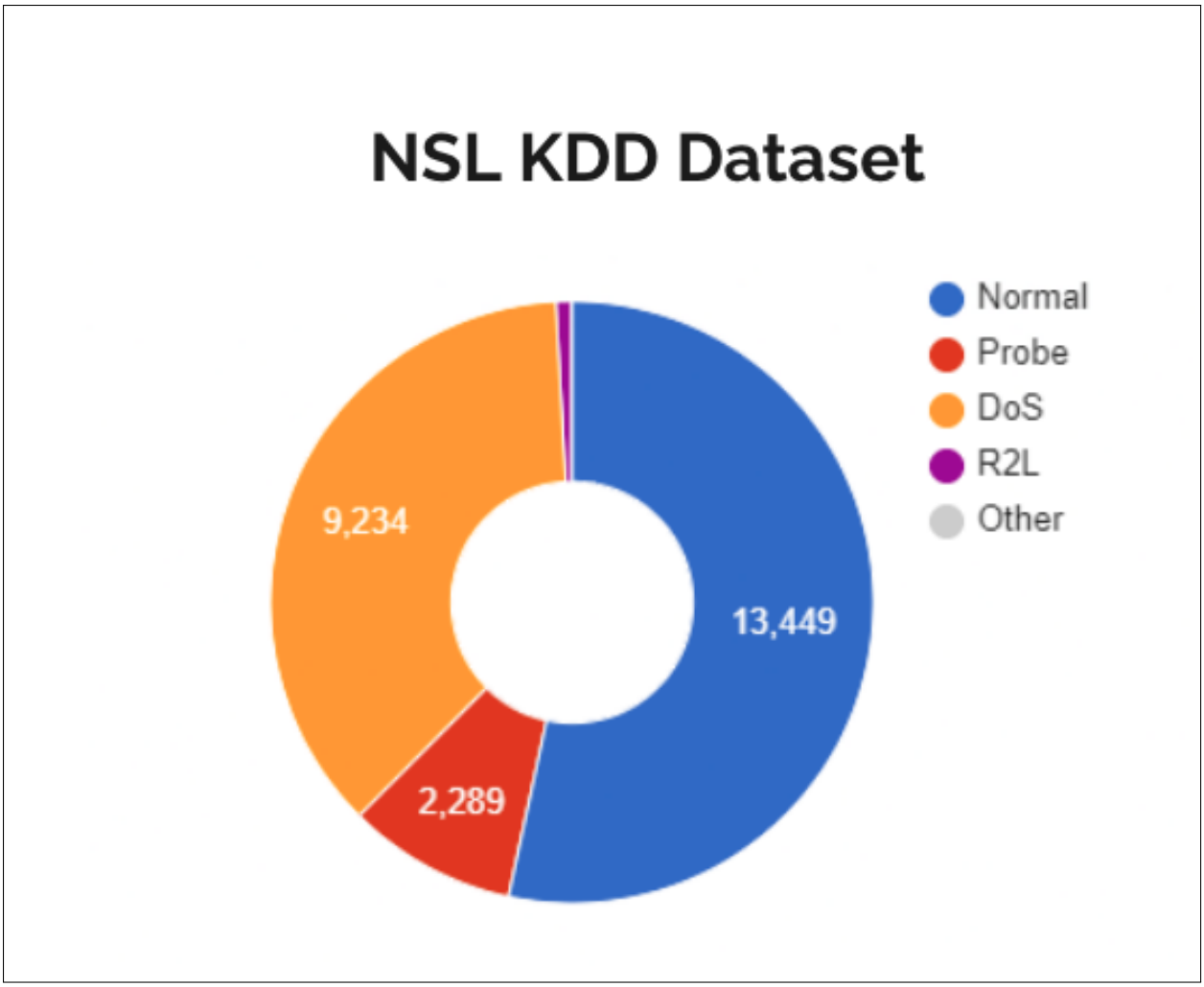

Figure 3.3: *Classwise Distribution in NSL-KDD*

1. **Categorical Encoding**
   The dataset has many columns with categorical data which is encoded to numbers, before it is used to fit and evaluate a model.

2. **Delete Duplicate values**
   Delete the sample's duplicate value, only keep one valid data.

3. **Detecting and removing Null Values**
   In the sample data, the sample size of missing values (Not a Number, NaN) and Infinite values(Inf) is small, which are deleted.

4. **Removal of unwanted features**
   In CSE-CIC-IDS2018,features such as "Timestamp", "Destination Address", "Source Address", "Source Port", etc. are deleted.If features "Init Bwd Win Byts" and features "Init Fwd Win Byts" have a value of -1, two check dimensions are added. The mark of -1 is 1. Otherwise, it is 0. In NSL-KDD, the OneHot encoder is used to complete this

conversion. For example, “TCP", “UDP" and “ICMP" are functions of three protocol types. After OneHot encoding, the features become binary vectors (1, 0, 0), (0, 1, 0), (0, 0, 1). The protocol type function can be divided into three categories, including 11 categories for flag function and 70 categories for service function. Therefore, the 41 dimensions initial feature vector becomes 122 dimensions.

5. **Standardize the data**
   In order to eliminate the dimensional influence between indicators and accelerate the gradient descent and model convergence, the data is standardized, that is, the method of obtaining Z-Score, so that the average value of each feature becomes 0 and the standard deviation becomes 1, converted to a standard normal distribution, which is related to the overall sample distribution, and each sample point can have an impact on standardization.

3.2.3 Experimental Parameters

The proposed method uses the Sklearn(machine learning framework) and completes the related experiments on the Nvidia DGX 800 machine. The machine learning algorithms use CPU calculations. The specific parameters are shown in table below.

Project	Properties
OS	Ubuntu 20.04.3 LTS
CPU	AMD EPYC 7742 64-Core Processor
Memory	18GB
Disk	503GB
Framework	SKlearn1.0.2

For the machine learning algorithms, the RandomForestClassifier, DecisionTreeClassifier and XGBoostClassifier experiments provided in Sklearn, are used. The specific parameters are given in the table below.

3.2.4 Evaluation Metrics

The Accuracy,Prediction, Recall and F1-Score is used, to evaluate the experimental classification model’s performance. Based on a classification model’s performance, there are four possible outcomes:

- True Positive(TP): Actually Positive and Predicted Positive

Classifier	Parameters
RandomForestClassifier	n_estimators=200, criterion='gini', min_samples_split=2, min_samples_leaf=1
DecisionTreeClassifier	criterion="gini", min_samples_split=2, min_samples_leaf=1
XGBClassifier	objective='multi:softmax', booster='gbtree', verbosity=0, silent=0, learning_rate=0.1, use_label_encoder=False

- True Negative(TN): Actually Negative and Predicted Negative
- False Positive(FP): Actually Negative and Predicted Positive
- False Negative(FN): Actually Positive and Predicted Negative

Using the above outcomes, the four metrics can be formulated as:

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

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

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

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Chapter 4

Methodologies

4.1 Ensemble of Imbalance Reduction Techniques

4.1.1 Steps followed while forming the ensemble of IRTs:

1. Splitting into training and test data
2. Applying each individual technique separately on the training dataset
3. Concatenating all the new training datasets thus generated
4. Removing all the duplicates from the concatenated dataset to obtain the final training dataset.

4.1.2 Various Ensembles used

1. **Ensemble of OverSampling Techniques(all):**
   SMOTE, BorderlineSMOTE, RandomOverSampling and ADASYN
2. **Ensemble of OverSampling Techniques(selected)**
   SMOTE and RandomOverSampling
3. **Ensemble of UnderSampling Techniques**
   TomekLinks, RandomUnderSampling, CLuster Centroid, Edited Nearest Neighbour, Instance Hardness Threshold

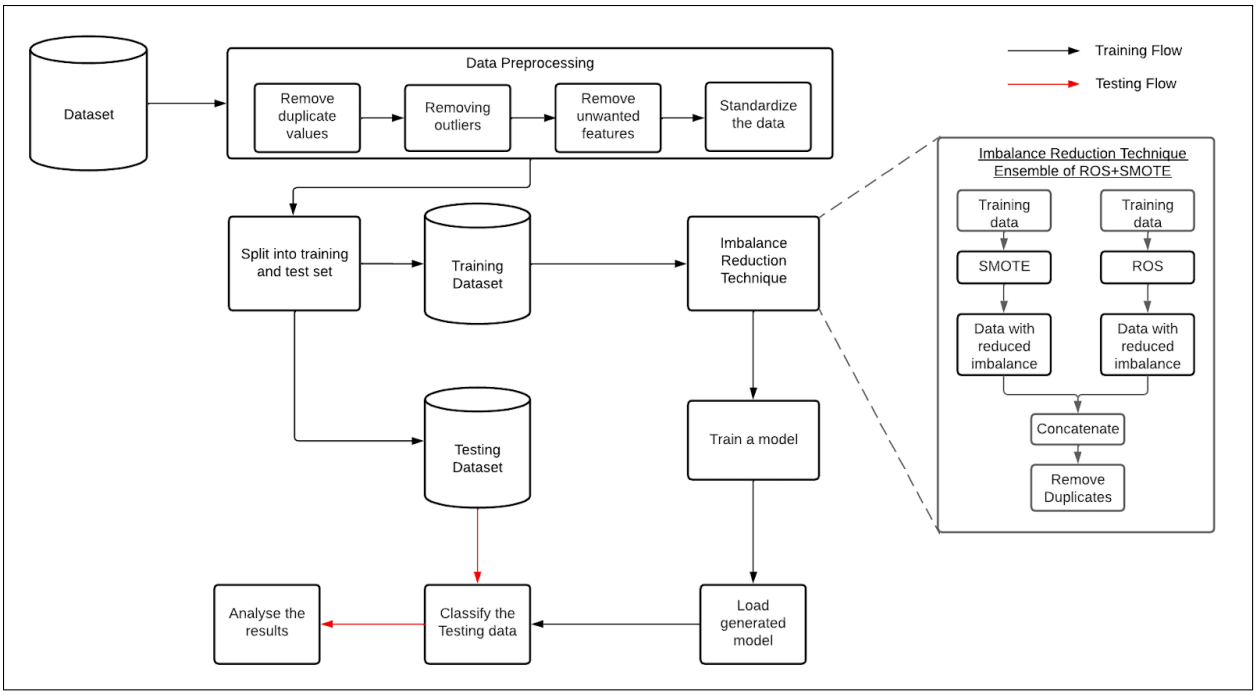

Figure 4.1: This is the *Ensemble System Architecture*. After finishing the preprocessing of the dataset, various Imbalance Reduction Techniques (either Oversampling or Undersampling) are applied on the dataset individually/in parallel. Upon completion, the datasets obtained as a result of these techniques are concatenated and the duplicate samples are removed. This resultant dataset is then used for training machine learning models.

4.2 Models Used

4.2.1 Random Forest Classifier

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

The fundamental concept behind random forest is a simple but powerful one: the wisdom of crowds.

The core unit of random forest classifiers is the decision tree. The decision tree is a hierarchical structure that is built using the features (or the independent variables) of a data set. Each node of the decision tree is split according to a measure associated with a subset of the features. The random forest is a collection of decision trees that are associated with a set of bootstrap samples that are generated from the original data set. The nodes are split based on the entropy (or Gini index) of a selected subset of the features. The subsets that are created from the original data set, using bootstrapping, are of the same size as the original data set.

4.2.2 Decision Tree

A Classification Tree is a supervised learning algorithm where the outcome variable is categorical/discrete. Internal nodes represent the dataset attributes, branches represent decision rules, and each leaf node provides the outcome category in this tree-structured classifier. The Decision Node and the Leaf Node are the two nodes of a Decision tree. Leaf nodes are the output of those decisions and do not contain any more branches, whereas Decision nodes are used to make any decision and have several branches.The decisions or tests are based on the characteristics of the given dataset. It's a graphical depiction for obtaining all feasible solutions to a problem/decision depending on certain parameters. It's termed a decision tree because, like a tree, it begins with the root node and grows from there.

4.2.3 XGBoost (eXtreme Gradient Boosting)

XGBoost is a tree based ensemble machine learning algorithm which is a scalable machine learning system for tree boosting. It is a parallel regression tree model that combines the idea of Boosting, which is improved based on gradient descent decision tree. Compared with the GBDT (Gradient Boosting Decision Tree) model, XGBoost overcomes the limited calculation speed and accuracy. XGBoost adds regularization to the original GBDT loss function to prevent the model from overfitting. The traditional GBDT performs a first-order Taylor expansion on the calculated loss function and takes the negative gradient value as the residual value of the current model. In contrast, XGBoost performs a second-order Taylor expansion to ensure the accuracy of the model. Moreover, XGBoost blocks and sorts each feature, making it possible to parallelize the calculation when looking for the best split point, which significantly accelerates the calculation speed.

Chapter 5

Results

The results of Accuracy, F1 Score, Precision Score and Recall are presented on each of the models : RandomForest, Decision Tree and XGBoost.

First, the result of the models without applying any imbalance methods to obtain the base values are presented. Further, the results of individual imbalance reduction methods are shown for better comparison.

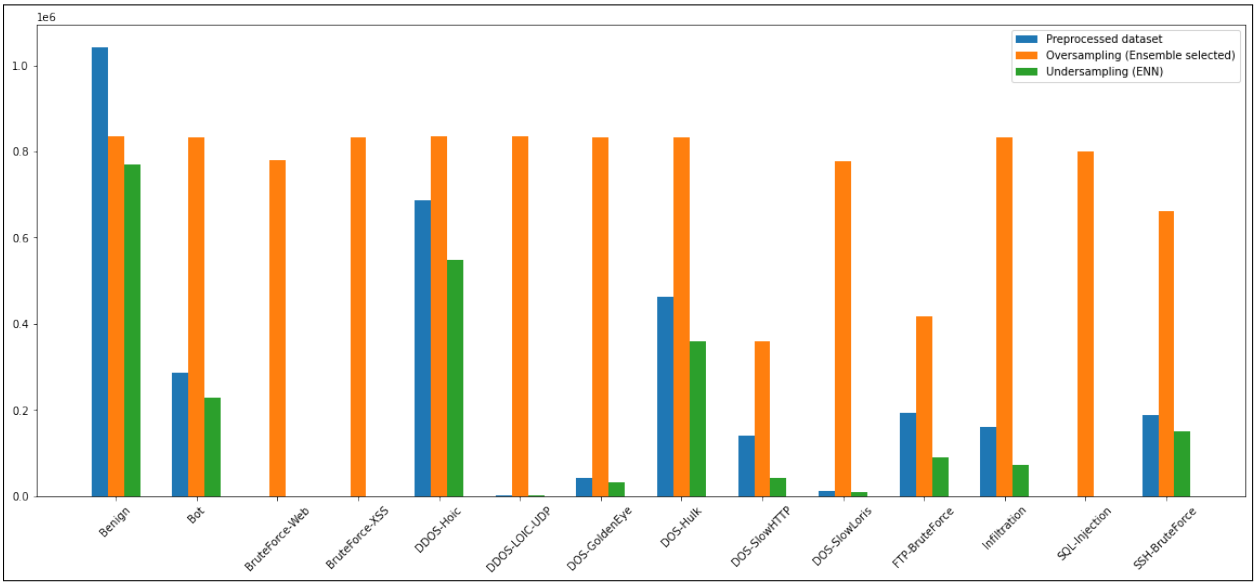

Figure 5.1: Diagram 4.1: This diagram gives a pictorial description of the unaltered dataset (preprocessed dataset: in blue) and how the dataset samples vary according to the respective imbalance reduction methods applied. The orange bar depicts the dataset resulting from the application of Oversampling Technique (Ensemble Selected) and the green bar is for the Undersampled dataset (ENN).

Ensemble of OverSampling Methods of SMOTE, ROS, BorderlineSMOTE and ADASYN

shows better accuracy for RandomForest, and a much better Precision in SVM , along with a overall better performance in XGBoost.

Similar is the case with, Ensemble of UnderSampling Methods which shows a better F1 score for RandomForest and a overall better performance in XGBoost.

After using practically all of the imbalance reduction approaches, there is a significant improvement in accuracy. It is found that Adasyn outperforms all other over sampling methods, while Random under sampling outperforms all other under sampling methods. These strategies enhance accuracy as well as the other metrics: F1 score, Precision, and Recall.

5.1 Random Forest

		NSL-KDD			
	Technique	Accuracy	F1 score	Precision	Recall
	Without Imbalance Reduction	0.7477	0.703	0.8157	0.7477
Over Sampling Methods (OS)	Smote	0.755	0.7172	0.8155	0.755
	Random Over Sampling	0.7394	0.6922	0.7855	0.7394
	Borderline Smote	0.7565	0.7151	0.8154	0.7565
	Adasyn	0.7733	0.7376	0.826	0.7733
	Ensemble OS (all)	0.7622	0.7241	0.8192	0.7622
	Ensemble OS (selected)	0.7533	0.7167	0.8148	0.7533
Under Sampling Methods (US)	Random Under Sampling	0.796	0.8001	0.8483	0.796
	Tomek Links	0.7416	0.6954	0.8083	0.7416
	Cluster Centroids	0.333	0.3126	0.5012	0.333
	Edited Nearest Neighbor	0.7389	0.691	0.8145	0.7389
	Instance Hardness Threshold	0.7541	0.7129	0.823	0.7541
	Ensemble US (all)	0.7658	0.725	0.8253	0.7658
OS+US	DSSTE	0.7499	0.7041	0.8073	0.7499

Table 5.1: The table shows the results obtained by training Random Forest model after different applying imbalance reduction techniques on **NSL-KDD** daatset. In the table **Ensemble OS (all)** is the to ensemble of all oversampling methods, **Ensemble OS (selected)** is the ensemble of SMOTE and ROS methods and **Ensemble US** is the ensemble of all undersampling methods.

		CSE-CIC-2018			
	Technique	Accuracy	F1 score	Precision	Recall
	Without Imbalance Reduction	0.9577	0.9575	0.9578	0.9577
Over Sampling Methods (OS)	Smote	0.9467	0.9484	0.9641	0.9467
	Random Over Sampling	0.9459	0.9478	0.964	0.9459
	Borderline Smote	0.9447	0.9444	0.9444	0.9447
	Adasyn	0.9438	0.9447	0.9617	0.9438
	Ensemble OS (all)	0.9421	0.9444	0.9572	0.9421
	Ensemble OS (selected)	0.9472	0.9488	0.964	0.9472
Under Sampling Methods (US)	Random Under Sampling	0.8934	0.9036	0.9388	0.8934
	Tomek Links	0.9575	0.9573	0.9577	0.9575
	Cluster Centroids	0.8411	0.8437	0.8701	0.8411
	Edited Nearest Neighbor	0.9563	0.9566	0.9583	0.9563
	Instance Hardness Threshold	0.9509	0.9507	0.9529	0.9509
	Ensemble US (all)	0.9566	0.9568	0.9582	0.9566
OS+US	DSSTE	0.9523	0.9523	0.9547	0.9523

Table 5.2: The table shows the results obtained by training Random Forest model after different applying imbalance reduction techniques on **CSE-CIC-IDS-2018** daatset. In the table **Ensemble OS (all)** is the to ensemble of all oversampling methods, **Ensemble OS (selected)** is the ensemble of SMOTE and ROS methods and **Ensemble US** is the ensemble of all undersampling methods.

These are the results obtained after training the NSL-KDD and CSE-CIC-2018 datasets for Random Forest classifier. For NSL-KDD dataset, random forest achieved the highest accuracy after applying Adasyn as over sampling method. Among the individual undersampling techniques, Random Under Sampling achieved highest accuracy. However, the performance improvement was very small compared to original dataset. The ensemble of all the Undersampling techniques has better accuracy

The imbalance reduction techniques which gave comparatively better results for CSE-CIC-2018 dataset, under the paradigm of Oversampling was the Ensemble of SMOTE and Random Oversampling. Amongst Undersampling Techniques, Edited Nearest Neighbours gave the best results, followed by Tomek Links and then Instance Hardness Threshold.

5.2 Decision Tree

		NSL-KDD			
	Technique	Accuracy	F1 score	Precision	Recall
	Without Imbalance Reduction	0.7729	0.7368	0.814	0.7729
Over Sampling Methods (OS)	Smote	0.7438	0.7059	0.7109	0.7438
	Random Over Sampling	0.7457	0.7051	0.7635	0.7457
	Borderline Smote	0.7521	0.7147	0.7322	0.7521
	Adasyn	0.7751	0.7334	0.8077	0.7751
	Ensemble OS (all)	0.7611	0.7167	0.7688	0.7611
	Ensemble OS (selected)	0.7587	0.7128	0.7284	0.7587
Under Sampling Methods (US)	Random Under Sampling	0.7296	0.7507	0.8139	0.7296
	Tomek Links	0.7409	0.7021	0.7024	0.7409
	Cluster Centroids	0.3405	0.3763	0.5418	0.3405
	Edited Nearest Neighbor	0.7698	0.7306	0.8137	0.7698
	Instance Hardness Threshold	0.7513	0.7159	0.8106	0.7513
	Ensemble US (all)	0.7642	0.7283	0.8102	0.7642
OS+US	DSSTE	0.7754	0.7374	0.823	0.7754

Table 5.3: The table shows the results obtained by training Decision Tree model after different applying imbalance reduction techniques on **NSL-KDD** dataset. In the table **Ensemble OS (all)** is the to ensemble of all oversampling methods, **Ensemble OS (selected)** is the ensemble of SMOTE and ROS methods and **Ensemble US** is the ensemble of all undersampling methods.

		CSE-CIC-2018			
	Technique	Accuracy	F1 score	Precision	Recall
	Without Imbalance Reduction	0.9536	0.9538	0.954	0.9536
Over Sampling Methods (OS)	Smote	0.952	0.9523	0.9533	0.952
	Random Over Sampling	0.9531	0.9532	0.9533	0.9531
	Borderline Smote	0.9452	0.9451	0.945	0.9452
	Adasyn	0.9423	0.9421	0.9422	0.9423
	Ensemble OS (all)	0.9496	0.9491	0.9549	0.9496
	Ensemble OS (selected)	0.9545	0.9547	0.9576	0.9545
Under Sampling Methods (US)	Random Under Sampling	0.8889	0.8958	0.9243	0.8889
	Tomek Links	0.9561	0.9563	0.9567	0.9561
	Cluster Centroids	0.7849	0.8009	0.8507	0.7849
	Edited Nearest Neighbor	0.9584	0.9585	0.9587	0.9584
	Instance Hardness Threshold	0.953	0.9529	0.9529	0.953
	Ensemble US (all)	0.9534	0.9536	0.954	0.9534
OS+US	DSSTE	0.9547	0.9547	0.9547	0.9547

Table 5.4: The table shows the results obtained by training Decision Tree model after different applying imbalance reduction techniques on **CSE-CIC-IDS-2018** daatset. In the table **Ensemble OS (all)** is the to ensemble of all oversampling methods, **Ensemble OS (selected)** is the ensemble of SMOTE and ROS methods and **Ensemble US** is the ensemble of all undersampling methods.

The results above show that, for NSL-KDD dataset, ADASYN has given best results. Only the Ensemble of all the oversampling methods – Random Oversampling, SMOTE, Borderline SMOTE and ADASYN has given results which are closer to those obtained without any imbalance reduction. Random Undersampling has given results better F1 score than all other Undersampling Techniques. The results from DSSTE have shown improvement than in the unaltered dataset.

The CSE-CIC-2018 dataset sees a trend here, which similar to Random Forest. Out of all the Oversampling Techniques, Ensemble of Random Oversampling and SMOTE have given best results. Edited Nearest Neighbour have provided overall better results, and Tomek Links has given slightly better metrics than the rest.

5.3 XGBoost

		NSL-KDD			
	Technique	Accuracy	F1 score	Precision	Recall
	Without Imbalance Reduction	0.764	0.7226	0.816	0.764
Over Sampling Methods (OS)	Smote	0.7813	0.7536	0.8279	0.7813
	Random Over Sampling	0.7645	0.7255	0.8189	0.7645
	Borderline Smote	0.7822	0.746	0.8257	0.7822
	Adasyn	0.7944	0.7691	0.8307	0.7944
	Ensemble OS (all)	0.7803	0.7566	0.8255	0.7803
	Ensemble OS (selected)	0.7758	0.7473	0.8264	0.7758
Under Sampling Methods (US)	Random Under Sampling	0.8147	0.8162	0.8514	0.8147
	Tomek Links	0.7607	0.7189	0.8139	0.7607
	Cluster Centroids	0.3322	0.291	0.4361	0.3322
	Edited Nearest Neighbor	0.751	0.7087	0.8108	0.751
	Instance Hardness Threshold	0.7576	0.7163	0.8161	0.7576
	Ensemble US (all)	0.7693	0.7282	0.8189	0.7693
OS+US	DSSTE	0.7611	0.7246	0.8227	0.7611

Table 5.5: The table shows the results obtained by training XGBoost model after different applying imbalance reduction techniques on **NSL-KDD** dataset. In the table **Ensemble OS (all)** is the to ensemble of all oversampling methods, **Ensemble OS (selected)** is the ensemble of SMOTE and ROS methods and **Ensemble US** is the ensemble of all undersampling methods.

		CSE-CIC-2018			
	Technique	Accuracy	F1 score	Precision	Recall
	Without Imbalance Reduction	0.9577	0.9575	0.9578	0.9577
Over Sampling Methods (OS)	Smote	0.9467	0.9484	0.9641	0.9467
	Random Over Sampling	0.9459	0.9478	0.964	0.9459
	Borderline Smote	0.9275	0.9296	0.9365	0.9275
	Adasyn	0.9438	0.9447	0.9617	0.9438
	Ensemble OS (all)	0.9421	0.9444	0.9572	0.9421
	Ensemble OS (selected)	0.9472	0.9488	0.964	0.9472
Under Sampling Methods (US)	Random Under Sampling	0.8934	0.9036	0.9388	0.8934
	Tomek Links	0.9575	0.9573	0.9577	0.9575
	Cluster Centroids	0.8411	0.8437	0.8701	0.8411
	Edited Nearest Neighbor	0.9563	0.9566	0.9583	0.9563
	Instance Hardness Threshold	0.9509	0.9507	0.9529	0.9509
	Ensemble US (all)	0.9566	0.9568	0.9582	0.9566
OS+US	DSSTE	0.9523	0.9523	0.9547	0.9523

Table 5.6: The table shows the results obtained by training Decision Tree model after different applying imbalance reduction techniques on **CSE-CIC-IDS-2018** daatset. In the table **Ensemble OS (all)** is the to ensemble of all oversampling methods, **Ensemble OS (selected)** is the ensemble of SMOTE and ROS methods and **Ensemble US** is the ensemble of all undersampling methods.

Random Undersamling has shown the best improvement in the results for the NSL-KDD dataset. There is a huge betterment in the F1 score under this technique. The remaining Undersampling techniques have shown some upswing in the metrics. Ensemble of all the Undersampling Techniques has the next best results. However, the results of all the Oversampling Techniques have shown amelioration. ADASYN has provided the best F1 score and accuracy. It is closely followed by Ensemble of all Oversampling Techniques and Borderline SMOTE.

Only Tomek Links has given results better than those from without any imbalance reduction for the CSE-CIC-IDS-2018 dataset. Apart from that, the F1 score of Edited Nearest Neighbours is the closest to the unaltered dataset. Both of these methods have outperformed DSSTE algorithm. Random Oversampling, SMOTE, and Ensemble of SMOTE and Random Oversampling have shown to give better results amongst the Oversampling techniques. These methods also have given the highest precision, better than that of unaltered dataset.

Chapter 6

Conclusion

After having gone through research papers studying various models applied on benchmark datasets i.e. CSE-CIC-IDS 2018 and NSL-KDD datasets, it was noticed that the literature gap with respect to the imbalance reduction techniques as mentioned above.

Following that, various techniques to reduce imbalance have been used and the results obtained using individual undersampling and oversampling imbalance reduction methods and their respective ensembles have been compared. Apart from these, the recently published DSSTE technique has also been used and its results are compared.

A novel way of imbalance reduction has been developed, which involves implementing various techniques for reducing imbalance and forming an ensemble of the resultant datasets. This dataset, formed as a result of the ensemble of Oversampling/Undersampling techniques is then further used for training machine learning models for Network Intrusion Detection Systems.

The results show that the use of imbalance reduction techniques has led to an improvement in the model accuracy, F1 score, recall and precision values.

For NSL-KDD dataset, through the experiments, it is clear that Adasyn has consistently given better results amongst the Oversampling Techniques applied for reducing imbalance in the dataset. It has outperformed the results obtained from unaltered dataset and the DSSTE algorithm. Out of the Undersampling Techniques, Random Undersampling is seen to have given the best results. Apart from this, the Ensemble of all the Undersampling Techniques has provided improvement in the metrics. Ensembles of all and the selected (Random Oversampling and SMOTE) oversampling methods have shown to give improvement in most cases. This implies that although ensembles help in enhancing the results, this need not always be the case.

Tomek Links has consistently provided better results for all the models trained using CSE-CIC-IDS-2018 dataset. Except for XGBoost, Edited Nearest Neighbours technique has vastly

outperformed all the other imbalance reduction method. The ensemble of Random Oversampling and SMOTE has been a very useful technique for reducing imbalance out of all other oversampling techniques. It has outperformed the DSSTE algorithm for all the models trained.

Through the experimentations it is concluded that not all techniques are suitable for all the datasets. It can vary from one dataset to another and the only way to know that is through trial and error. No single imbalance reduction Technique outperformed all others for all datasets and all models. Ensembles of certain techniques did prove to show better results than the rest in some cases. Depending upon the permutation of techniques used for the ensemble, the results have varied. However, in all scenarios, ensemble have provided comparable, if not better, results.

Chapter 7

Future Scope

There is the possibility of trying out various combinations of Ensembles of imbalance reduction techniques and see which ones provide better results.

Training machine learning models apart from those used here for these datasets and experimenting with the imbalance reduction techniques.

DSSTE is the only technique which combines both, Oversampling and Undersampling. Other combinations of techniques can be tried.

Use trained models for intrusion detection in real world scenarios to know how efficient the models really are.

Bibliography

[1] Muhammad Shakil Pervez and Dewan Md Farid. "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs". In: *SKIMA 2014 - 8th International Conference on Software, Knowledge, Information Management and Applications* (Apr. 2014). DOI: 10.1109/SKIMA.2014.7083539.

[2] L Dhanabal and S P Shantharajah. "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms". In: *International Journal of Advanced Research in Computer and Communication Engineering* 4 (2015). DOI: 10.17148/IJARCCE.2015.4696.

[3] Mohammad Reza Parsaei, Samaneh Miri Rostami, and Reza Javidan. "A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset". In: *IJACSA) International Journal of Advanced Computer Science and Applications* 7 (6 2016). URL: www.ijacsa.thesai.org.

[4] Preeti Aggarwal and Sudhir Kumar Sharma. "Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection". In: *Procedia Computer Science* 57 (2015), pp. 842–851. ISSN: 18770509. DOI: 10.1016/j.procs.2015.07.490.

[5] Richard Zuech, John Hancock, and Taghi M. Khoshgoftaar. "Detecting web attacks using random undersampling and ensemble learners". In: *Journal of Big Data* 8 (1 Dec. 2021). ISSN: 21961115. DOI: 10.1186/s40537-021-00460-8.

[6] Joffrey L. Leevy et al. "Detecting cybersecurity attacks across different network features and learners". In: *Journal of Big Data* 8 (1 Dec. 2021). ISSN: 21961115. DOI: 10.1186/s40537-021-00426-w.

[7] Sugandh Seth, Gurvinder Singh, and Kuljit Kaur Chahal. "A novel time efficient learning-based approach for smart intrusion detection system". In: *Journal of Big Data* 8 (1 Dec. 2021). ISSN: 21961115. DOI: 10.1186/s40537-021-00498-8.

[8] Laurens D'hooge et al. "Inter-dataset generalization strength of supervised machine learning methods for intrusion detection". In: *Journal of Information Security and Applications* 54 (Oct. 2020). ISSN: 22142126. DOI: 10.1016/j.jisa.2020.102564.

[9] Xu Kui Li et al. "Building Auto-Encoder Intrusion Detection System based on random forest feature selection". In: *Computers and Security* 95 (Aug. 2020). ISSN: 01674048. DOI: 10.1016/j.cose.2020.101851.

[10] Miel Verkerken et al. "Towards Model Generalization for Intrusion Detection: Unsupervised Machine Learning Techniques". In: *Journal of Network and Systems Management* 30 (1 Jan. 2022). ISSN: 15737705. DOI: 10.1007/s10922-021-09615-7.

[11] Jiyeon Kim et al. "CNN-based network intrusion detection against denial-of-service attacks". In: *Electronics (Switzerland)* 9 (6 June 2020), pp. 1–21. ISSN: 20799292. DOI: 10.3390/electronics9060916.

[12] Shahab Shamshirband et al. "Computational intelligence intrusion detection techniques in mobile cloud computing environments: Review, taxonomy, and open research issues". In: *Journal of Information Security and Applications* 55 (Dec. 2020). ISSN: 22142126. DOI: 10.1016/j.jisa.2020.102582.

[13] Mohamed Amine Ferrag et al. "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study". In: *Journal of Information Security and Applications* 50 (Feb. 2020). ISSN: 22142126. DOI: 10.1016/j.jisa.2019.102419.

[14] Peng Lin, Kejiang Ye, and Cheng Zhong Xu. "Dynamic network anomaly detection system by using deep learning techniques". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11513 LNCS (2019), pp. 161–176. ISSN: 16113349. DOI: 10.1007/978-3-030-23502-4_12.

[15] Sireesha Rodda and Uma Shankar Rao Erothi. "Class imbalance problem in the Network Intrusion Detection Systems". In: *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016* (Nov. 2016), pp. 2685–2688. DOI: 10.1109/ICEEOT.2016.7755181.

[16] David A. Cieslak, Nitesh V. Chawla, and Aaron Striegel. "Combating imbalance in network intrusion datasets". In: *2006 IEEE International Conference on Granular Computing* (2006), pp. 732–737. DOI: 10.1109/grc.2006.1635905.

[17] Piyasak Jeatrakul, Kok Wai Wong, and Chun Che Fung. "Classification of Imbalanced Data by Combining the Complementary Neural Network and SMOTE Algorithm". In: *LNCS* 6444 (2010), pp. 152–159.

[18] Binghao Yan and Guodong Han. "LA-GRU: Building Combined Intrusion Detection Model Based on Imbalanced Learning and Gated Recurrent Unit Neural Network". In: *Security and Communication Networks* 2018 (2018). ISSN: 19390122. DOI: 10.1155/2018/6026878.

[19] Lan Liu et al. "Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning". In: *IEEE Access* 9 (2021), pp. 7550–7563. ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3048198.

[20] Dennis L Wilson. "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data". In: (2 1972).

[21] Nitesh V Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357.

[22] Institute of Electrical and Electronics Engineers. “Neural Networks, 2008, IJCNN 2008, (IEEE World Congress on Computational Intelligence), IEEE International Joint Conference on : date, 1-8 June 2008.” In: ().