Department of Electrical and Computer Engineering
North South University

A novel Approach to LUFlow Intrusion Detection System Using Machine Learning Method

Taslima Akter **ID 1721599042**

MD Asif Mahmud **ID 1712841642**

Intisha Islam **ID 1711854642**

Faculty Advisor:

Dr Mohammad Monirujjaman Khan

Associate Professor

Electrical and Computer Engineering Department

Fall, 2021

DECLARATION

This is to certify that this Project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

Students' name & Signature

1. _____

2. _____

3. _____

APPROVAL

The Senior design project entitled "**A novel Approach to LUFlow Intrusion Detection System Using Machine Learning Method**" by **Taslima Akter (ID#1721599042), .MD Asif Mahmud (ID#1712841642), Intisha Islam (ID #1711854642)** is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering and has been accepted as satisfactory.

Supervisor's Signature

Dr. Mohammad Monirujjaman Khan
Associate Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

Department Chair'sSignature

Dr. Rajesh Palit
Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh.

ACKNOWLEDGMENT

First of all, we wish to express our gratitude to the Almighty for giving us the strength to perform our responsibilities and complete the report.

The research program is very helpful to bridge the gap between the theoretical knowledge and real life experience as part of Bachelor of Science (BSc) program. This report has been designed to have a practical experience through the theoretical understanding.

We also acknowledge our profound sense of gratitude to all the teachers who have been instrumental for providing us the technical knowledge and moral support to complete the project with full understanding.

It is imperative to show our appreciation for our honorable faculty member Dr. Mohammad Monirujjaman Khan for his undivided attention and help to achieve this milestone. Also, our gratefulness is divine to the North South University, ECE department for providing us a course such as CSE 498r in which we could really work on this project and materialize it the way we have dreamt of.

We thank our friends and family for their moral support to carve out this project and always offer their support.

ABSTRACT

Abstract: Recent years have seen significant growth in cybersecurity threats, and the strategies employed by attackers are continuing to improve and become more inventive. Furthermore, the complexity of most datasets, as well as the frequent occurrence of uneven class distributions, emphasizes the need for further study. The goal of this study is to use a variety of strategies for dealing with unbalanced datasets to create an effective intrusion detection system using the most recent LUFlow intrusion detection dataset. LUFlow is a flow-based network intrusion detection data collection that includes a reliable ground truth based on harmful behavior correlation. The efficacy of sampling strategies on LUFlow is thoroughly examined and empirically tested using machine learning classifiers such as Random Forest, Decision Tree, Gradient Boosting, and Ada boost. When dealing with the unbalanced class distribution with fewer samples, the suggested system was able to identify attacks with up to 100% F1-score, making it more practical in real-time data fusion situations that target data categorization. The study of the datasets used for training and testing in the detection model is also important since greater dataset quality may help advance offline intrusion detection. Benchmark datasets such as KDD99 and NSL-KDD Cup 99 are obsolete and have significant flaws, making them unfit for testing anomaly-based network intrusion detection systems. By combining the most popular feature selection methods and classifiers, such as Random Forest Classification (RF), Decision Tree (DT) Classification, Ada Boost (AB), and Gradient Boosting, the CIDDS-001 dataset has an ideal union of feature selection methodologies and classifiers (GB). Gradient boosting, Ada boost, and the decision tree classifier were found to be 99.72 percent, 99.34 percent, and 99.87 percent accurate, respectively.

Table of Content

List of Figures

Figure Page No.

CHAPTER 1

Introduction

Introduction

The concept of anomaly detection was first presented in 1987 [1]. The basic idea behind this method is to characterize the network and/or system's behavior, which is then compared to normal behavior. Accepting it or triggering the alert management system for additional examination will be the outcome. Many datasets derived from fused data for classification purposes have an unbalanced data distribution because the classes are not spread evenly. In such circumstances, practically all of the samples belong to one class, whereas the other class has fewer samples (typically the more important class). The CIDDS001 intrusion dataset is an example of this. When the number of samples for each class is almost equal, the majority of machine learning methods function well. The classifier prediction will be biased towards the majority class when the number of samples of one class is considerably more than the other and hence will not represent genuine findings. This research focuses on the CIDDS-001 intrusion dataset, and this research analyzes this topic and the efficacy of addressing unbalanced class distribution techniques. Tama et al. [2] employed deep neural networks mixed with the grid search approach to categorize assaults in Internet of Things (IoT) settings. In the classification phase, Tama and Rhee [2] used 10-fold cross-validation (10-FCV) and repeated cross-validation of 5 repetitions each using 2-FCV techniques. The authors divided the data into five sub-samples and used 2-FCV to verify them. The sub-samples were picked at random. As a result, the method may continually employ the same samples from the dataset's majority class. In this study, we use sampling strategies that consider the class distribution of instances. This is despite the fact that Tama and Rhee's validation procedures [2] had no effect on the performance of their suggested strategy. R. Thornton [3] presented a model for detecting unknown threats that combines ML and DL techniques. The authors reviewed several ML and DL strategies that fall under the categories of supervised and unsupervised learning. To attain great accuracy, however, a suitable standardized data collection must be provided. According to the various peer-reviewed papers for research in 2017, journals and conferences related to ML and DL reached an all-time high. This trend is expected to continue, with an increase in the number of publications on ML and DL-based intrusion detection systems. According to the survey, the most frequently utilized techniques for NIDS are DL and ANN. While KDD199 [4] and NSL KDD [5] are the most often

used datasets for NIDS assessment. The studies' authors [6-7] devised a paradigm in which numerous data sources, such as packet flow, logs, and sessions, are aggregated and fed into ML algorithms. They have also claimed that the world is moving toward artificial intelligence (AI), with AI being used in conjunction with automated technologies like NIDS. Signature-based intrusion detection systems are useless in the face of today's threats. The authors describe a method for intrusion detection in the study [8] that makes use of the discriminating features of genetic algorithms. KDD CUP 99 and UNSW-NBI5 datasets were used in experiments and calculations to show that the system worked well. Yi et al. discussed the detection challenges associated with flooding assaults [9-10]. To saturate the connection capacity, an intruder delivers massive packets of attack data as a route request. The author suggested neighbor suppression as a general protection mechanism for MANETs [11] against flooding threats. DoS attacks on wireless networks [12-14] may be quite damaging. He evaluated its efficacy. Yi et al. provide a cross-layer adaptive technique [15] for detecting black and gray hole assaults. Additionally, finite state machine detection [16], distributed intrusion detection [17], wireless intrusion prevention [18], multiagent cooperative intrusion response [19], and green firewall [20] are also IDS technologies. In our experiments (DT), we used supervised machine learning algorithms such as Random Forest, Adaboost, Gradient Boosting, and Decision Tree. To establish the relevance of each feature in the LUFlow dataset's intrusion detection, we utilize all machine learning methods. In this study, we used four machine learning algorithms, and they were all more than 98 percent accurate. Using the decision tree classifier and the Random Forest classifier, we achieved maximum accuracy of 99.87 percent and 99.92 percent, respectively. As a consequence, we were able to close the gap in our research's accuracy. Adopting these four models has the benefit of facilitating comparisons. These comparisons aid us in identifying the most accurate model. By simply using our technology, any business can identify infiltration. It will assist any business in preserving a significant amount of personal information, privacy, and money.

CHAPTER 2

Methodology

2.1 Introduction

Here we will talk about the motivations for which we thought of developing the system. We will also cover in this chapter the goal of this study is to use a variety of strategies for dealing with unbalanced datasets to create an effective intrusion detection system using the most recent LUFlow intrusion detection dataset. LUFlow is a flow-based network intrusion detection data collection that includes a reliable ground truth based on harmful behavior correlationMotivation towards our project.

2.2 Motivation towards our project

A variety of data science applications have compared and employed machine learning algorithms for analysis. This research-based project's primary driving goal was to investigate the feature selection techniques, data preparation, and processing techniques used in machine learning training models. Additionally, the goal of machine learning as a whole is to create a suitable computer system to identify cervical cancer and classify them using modern information. Using free online resources, the dataset wascreated [20]. The suggested approach begins by loading and extracting images and labels from raw datasets, followed by preprocessing and augmentation steps after the research and test sets have been separated. After this method is to characterize the network and/or system's behavior, which is then compared to normal behavior. Accepting it or triggering the alert management system for additional examination will be the outcome. Many datasets derived from fused data for classification purposes have an unbalanced data distribution because the classes are not spread evenly. In such circumstances, practically all of the samples belong to one class .

2.3 Summary

This chapter gave readers an understanding of the driving forces behind our project, which attempts to is to use a variety of strategies for dealing with unbalanced datasets to create an effective intrusion detection system using the most recent LUFlow intrusion detection dataset.

CHAPTER 3
Design results and analysis

3.1 Introduction

The many systems that are currently available on the market are covered in this chapter. We also pay attention to the flaws in the current system, and a convincing argument will be made for why our system is the best option right now.

3.2 Related Work to our project

The investigation of writing gives a speedy outline of the flow of research on calculations for any datasets derived from merged data for classification purposes have uneven distribution of data because the classes are not evenly distributed. In such situations, virtually all samples belong to one class, with fewer samples in other classes (usually the more important classes). Attack record CIDDS001 is an example. Most machine learning methods work well when the number of examples in each class is roughly the same. A classifier's predictions are biased toward the majority class when one class has significantly more samples than others, and thus is not representative of the true outcome. This study focuses on the CIDDS-001 intrusion dataset to analyze this problem and the effectiveness of handling imbalanced class distribution techniques. Tama et al used a deep neural network combined with a grid search approach to classify attacks in Internet of Things (IoT) environments. For the classification phase, Tama and Rhee used 10-fold cross-validation (10-FCV) and repeated cross-validation of five replicates each using the 2-FCV technique. The author split the data into her five subsamples and validated them using 2-FCV. Subsamples were randomly selected. As a result, the method can continuously use the same samples from the majority class of the data

set. In this study, we use a sampling strategy that considers the class distribution of instances. This is despite the fact that Tama and Rhee's validation procedure did not affect the performance of the proposed strategy. R. Thornton presented an unknown threat detection model that combines ML and DL techniques. The authors have identified several ML and DL strategies that fall into the categories of supervised and unsupervised learning. However, to achieve high accuracy, it is necessary to provide adequate standardized data collection. According to various peer-reviewed research papers in 2017, journals and conferences related to ML and DL reached a record high. This trend is expected to continue as the number of publications on ML- and DL-based intrusion detection systems increases. According to research, the most commonly used techniques in NIDS are DL and ANN. On the other hand, KDD199] and NSL KDD are the most commonly used data sets for NIDS evaluation. The authors of studies developed a paradigm in which numerous data sources such as packet flows, logs, and sessions are aggregated and fed to ML algorithms. They also argue that the world is moving towards artificial intelligence (AI), using AI in combination with automation techniques such as his NIDS. Signature-based attack detection systems are useless in the face of today's threats. In a study the authors describe a method of intrusion detection that takes advantage of the distinguishing features of genetic algorithms. We used the KDD CUP 99 and UNSW-NBI5 datasets for experiments and calculations to show that the system works well. Lee et al. We discussed the detection challenges associated with flood attacks. To saturate the link capacity, the intruder distributes large numbers of attack data packets as route requests. The authors proposed neighbor suppression as a general protection mechanism for MANET against flood threats. DoS attacks against wireless networks can cause considerable damage. he appreciated its effectiveness. Lee et al. We provide a cross-layer adaptive technique for

detecting attacks on black holes and gray holes. In addition, Finite State Machine Detection], Distributed Intrusion Detection, Wireless Intrusion Prevention, Multi-Agent Cooperative Intrusion Response, and Green Firewall are also IDS technologies. Our experiments (DT) used supervised machine learning algorithms such as Random Forest, Adaboost, Gradient Boosting, and Decision Trees. We use all machine learning techniques to determine the relevance of each feature in attack detection in the LUFlow dataset. The study used four machine learning algorithms, all of which were 98% or better. Maximum accuracies of 99.87 percent and 99.92 percent were achieved using a decision tree classifier and a random forest classifier, respectively. This allowed us to close a gap in research accuracy. Adopting these four models has the advantage of facilitating comparisons. These comparisons help identify the most accurate model. Any organization can detect intrusions simply by using our technology. It helps any company to retain large amounts of personal information, privacy and money.

CHAPTER 4

TECHNICAL DESIGN

4.1 Introduction

This section contains a detailed description of the dataset, a schematic of the proposed system, as well as schematics of several algorithms and evaluation matrices, as well as information on the study's technique and methodology.

4.2 Technical Design

The suggested system's block diagram is shown in Figure 1.

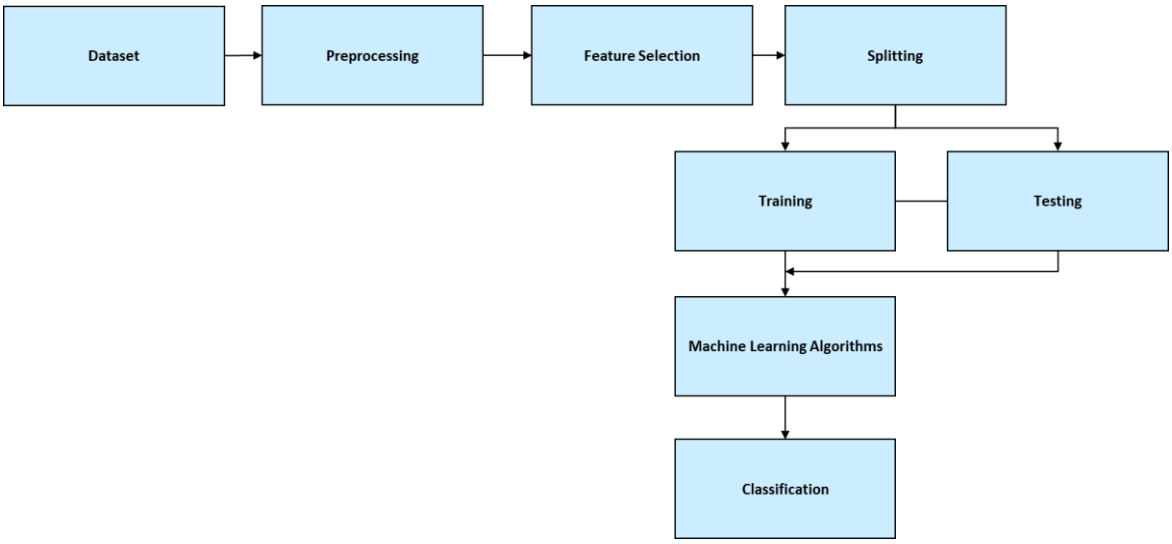

Figure 1: Block Diagram of the Proposed System

After the preparation step is complete, the data must be accessible to develop a model. You'll need a preprocessed dataset and machine learning methods to create the model. The research makes use of the Decision Tree (DT), Random Forest (RF), Adaboost (AB), and Gradient Boosting (GB) Classifiers.

4.2 Summary

As was already noted, the technical design has given us a clear understanding of how our system is functioning. We can therefore understand how our system is run by taking into account thedata flow diagram above.

CHAPTER 5

Experimental Analysis

5.1 Introduction

LUFlow is a labeled flow-based dataset that was published in 2017 in an OpenStack cloud environment [21]. This dataset supports unidirectional data packets and was developed using an emulation of a small company setting that comprised a web server, an e-mail server, and sets of variant clients.

5.2 data set

Figure 2 shows the total number of attacks and normal data.

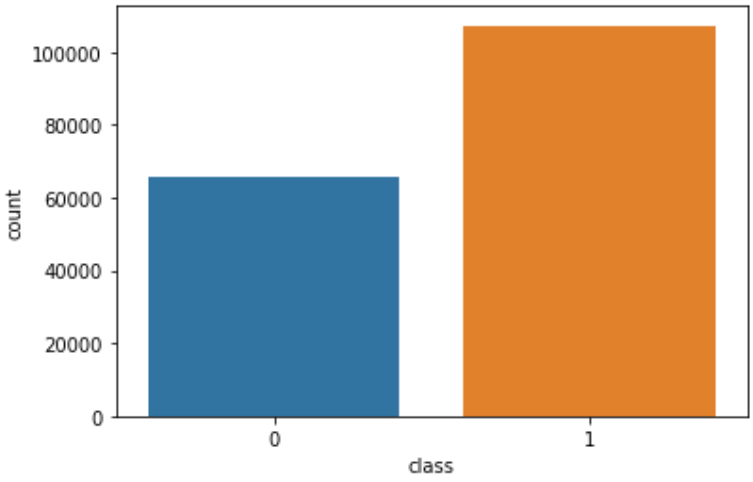

Figure 2: Total number of Attacks and Normal Data

The utilized dataset contains 65494 normal records and 107344 assault records.

5.3 Data Preprocessing

Preprocessing of data is the method of preparing raw data in such a manner that it may be utilized directly by an ML model. It is the first and most critical step in constructing ML models. For ML models, missing values, null values, and the wrong format for real-world data are rather rare occurrences. The accuracy and efficiency of a machine learning model may be increased by doing data preparation operations such as cleaning and preparing the data for use in the model. Our machine learning model may suffer significantly if our dataset contains missing data. As a consequence, it is critical to address the missing

values as well as fill in the null values in the dataset. The dataset is examined for null and missing values. Because our dataset has no missing or null values, it will remain unchanged. In order to improve accuracy while also increasing efficiency, the dataset is divided into two segments, which are referred to as training and test segments, and which have an 80/20 split ratio. Following the split, the classifier is constructed using a number of classifiers to improve its accuracy. The Decision Tree classifier (DT), Gradient Boosting Model (GB), Random Forest classifier (RF), and AdaBoost (AB) are the methods for classification used in this research.

5.4 Proposed Algorithms

Intrusion detection algorithms developed by Coburg are at the foundation of network security. Apart from discovering abnormalities, network abnormality detection systems monitor and analyses network events on a continual basis. The researchers evaluated four machine learning algorithms for identifying abnormalities using a publicly accessible dataset. Specifics include the following:

- Gradient Boosting
- Decision Tree
- Random Forest
- Ada Boost

5.4.1 Gradient Boosting

The gradient boosting approach, abbreviated as GBA, is a very effective and popular ML model. It is a technique in which each predictor attempts to improve on its previous performance by minimizing mistakes. However, the remarkable concept underlying gradient boosting is that, rather than fitting a

classifier to the data on every step, it adopts a new model to the regression produced by the preceding predictor. A block diagram of GB classifiers is shown in Figure 3.

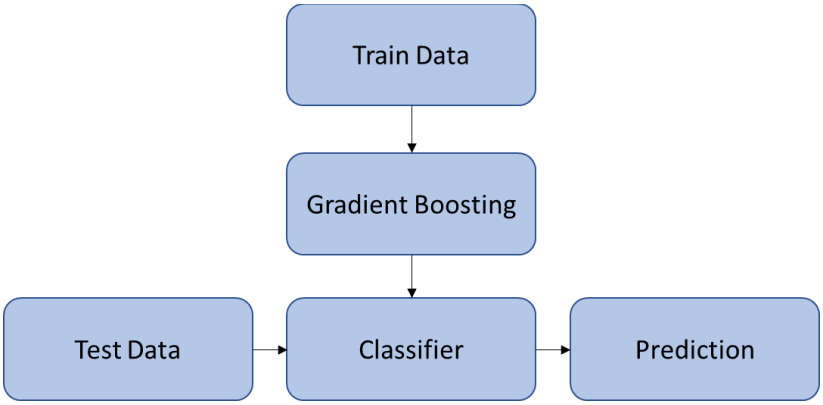

Figure 3: Diagram of Gradient boosting Classifier

Preprocessing of data is not a prerequisite in GB. That is why it is capable of dealing with missing data. This approach is very versatile and may be used to enhance a variety of different loss functions.

5.4.2 Decision Tree

Decision tree algorithms are heavily used in machine learning to handle classification and regression issues. A root node generates two categories of nodes, which are the internal node and the leaf node. The first node, internal, is characterized as decision-makers due to their various branches, while leaf nodes provide output due to their lack of further branches. As a result, it configures the structure of a tree. The decision tree classifier's fundamental architecture is shown in Figure 4.

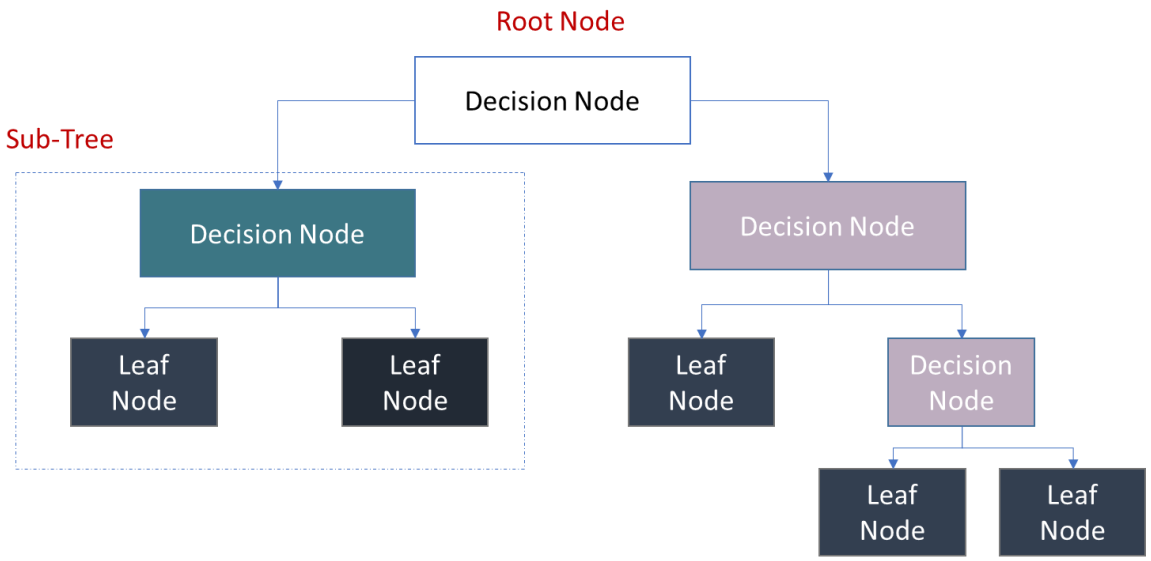

Figure 4: Diagram of Decision Tree Classifier

Data cleansing is the furthest thing from our minds when it comes to DT.It's simple to understand since it's shaped like a tree. The DT is straightforward to describe since it corresponds to the steps that a person goes through while selecting a real-world decision.

5.4.3 Random Forest

The Random Forest Classifier was utilized as the first algorithm. It comprises several independent trees, referred to as decision trees in RF, that is finally trained using training sample data. Following that, the outcomes from all these trees are subjected to a voting procedure in order to generate estimates. The ultimate outcome is determined by the majority of votes using an RF classifier. Figure 5 illustrates the RF diagram.

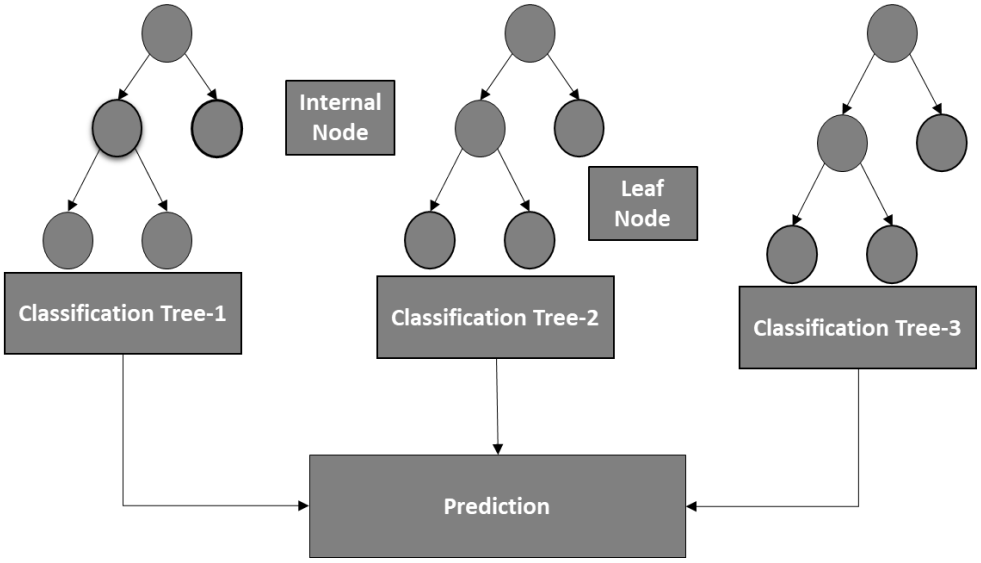

Figure 5: Diagram of Random Forest Classifier

Due to the fact that RF requires less training time than other algorithms, it will help us get better outcomes. The size of the dataset has little effect on the accuracy since it produces great accuracy on larger datasets. The absence of large volumes of data has no influence on accuracy. It also works well since the standard hyperparameters it uses generally convey reasonable expectations. Acknowledging the hyperparameters is crucial since there are so few. The primary issue that commonly arises in machine learning is overfitting. However, it happens less often in the random forest. Sufficient trees assist the classifier in avoiding overfitting.

5.4.4 Ada Boost Classifier

A strengthening strategy used in machine learning, the AdaBoost algorithm, is known as a prediction model. It is during this procedure that each instance is assigned a weight, with greater weights being assigned to instances that were misidentified. The subsequent learners, with the exception of the first, are developed from the prior ones. Simply said, poor learners, develop into strong learners. The block diagram of the ada boost model is shown in Figure 6.

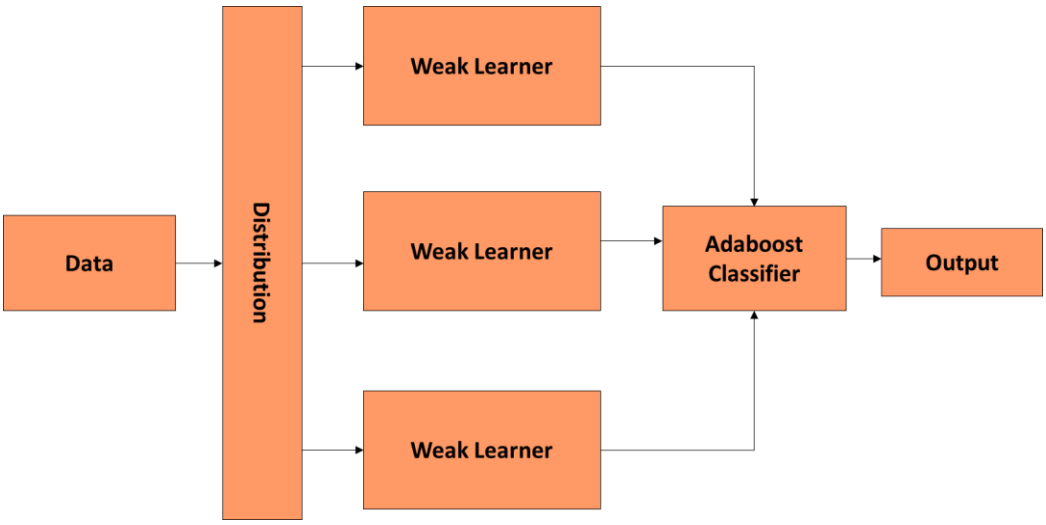

Figure 6: Diagram of Ada boost Classifier

Figure 6 illustrates how the initial model is constructed and how the algorithm identifies faults in the first model. The improperly categorized record is utilized as an input for the subsequent model. This procedure is continued until the condition stated is satisfied. As seen in Figure 6, three models are created by including the mistakes from the preceding model. This is the mechanism through which boosting operates.

5.5 Evaluation Matrix

The evaluation matrix is a statistic that indicates how well machine learning algorithms perform when confronted with a confusion matrix. The confusion matrix will be used to assess the whole collection of models. The confusion matrix illustrates how often our models produce accurate and inaccurate guesses.

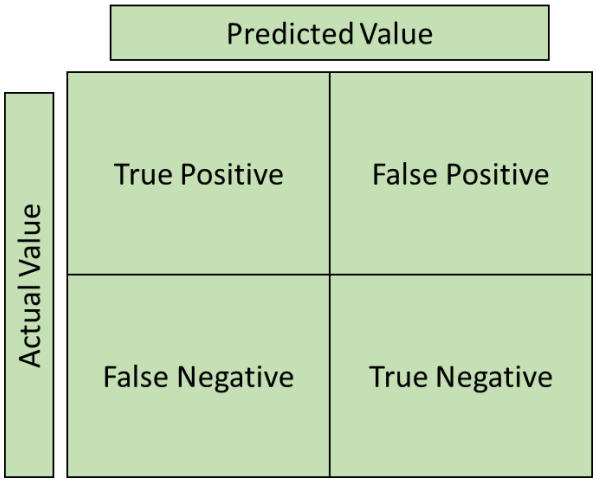

Figure 7: Block Diagram of Evaluation Matrix

As seen in Figure 7, false positives and negatives will be allocated to values that are poorly protected, whilst real positives and negatives will be assigned to effectively predicted values. After combining all estimated values in the matrix, the accuracy, precision-recall trade-off, and accuracy-recall trade-off were all examined and computed to assess the algorithm's performance.

CHAPTER 6

Result analysis

6.1 Introduction

The expected results and calculated performance of the model are shown in the confusion matrix. There were 51,810 correct guesses and 42 incorrect guesses.

6.1.1 Gradient Boosting

Figure 8 illustrates the Gradient boosting algorithm's classification accuracy.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19539
1	1.00	1.00	1.00	32313
accuracy			1.00	51852
macro avg	1.00	1.00	1.00	51852
weighted avg	1.00	1.00	1.00	51852

Figure 8: Classification accuracy of Gradient boosting classifier

The accuracy of the gradient boosting classifier was 99.72%. Both normal and intrusion data prediction have an F1 score of 100%. Both of them have perfect precision. The gradient boosting algorithm's confusion matrix is shown in Figure 9.

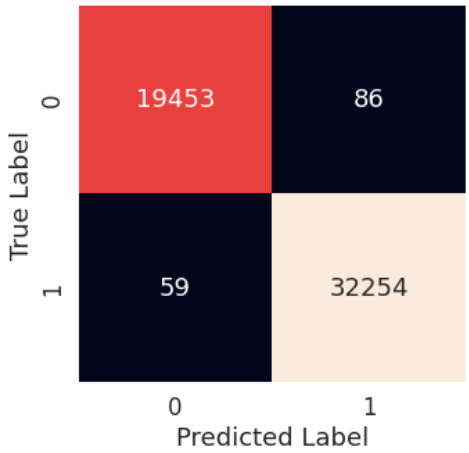

Figure 9: Confusion matrix of gradient boosting classifier

The figure shows the forecast provided by the gradient boosting model. The anticipated outcome and the model's calculated performance are shown in the confusion matrix. 51707 of the predictions were right, while just 145 were wrong.

6.1.2 Decision Tree

Figure 10 depicts the classification accuracy of the decision tree method.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19539
1	1.00	1.00	1.00	32313
accuracy			1.00	51852
macro avg	1.00	1.00	1.00	51852
weighted avg	1.00	1.00	1.00	51852

Figure 10: Classification accuracy of a decision tree classifier

The accuracy of the decision tree algorithm was 99.87 percent. For both normal and intrusion data prediction, the F1 score is 100 percent. For both 0 and 1, precision is 100 percent. Also, for both 0 and 1, recall is 100 percent. The decision tree algorithm's confusion matrix is shown in Figure 11.

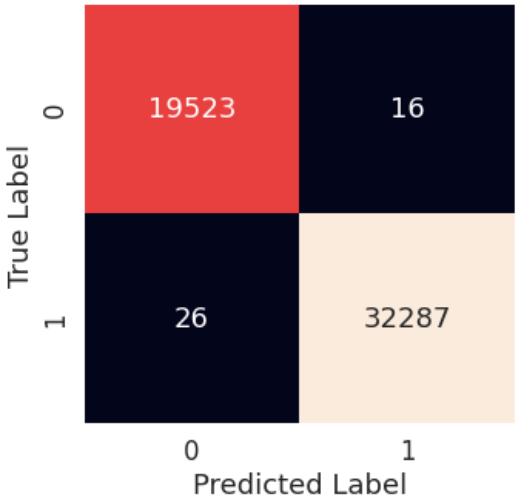

Figure 11: Confusion matrix of a decision tree classifier

Figure 11 shows the forecast provided by the decision tree model. The anticipated outcome and the model's calculated performance are shown in the confusion matrix. There were 51810 accurate and 42 erroneous guesses.

6.1.3 Random Forest

Figure 12 illustrates the Random forest algorithm's classification accuracy.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19539
1	1.00	1.00	1.00	32313
accuracy			1.00	51852
macro avg	1.00	1.00	1.00	51852
weighted avg	1.00	1.00	1.00	51852

Figure 12: Classification accuracy of Random forest classifier

The accuracy of the RF algorithm was 99.92 percent. For both normal and intrusion data prediction, the F1 score is 100 percent. For both 0 and 1, precision is 100 percent. Also, for both 0 and 1, recall is 100 percent. The RF algorithm's confusion matrix is shown in Figure 13.

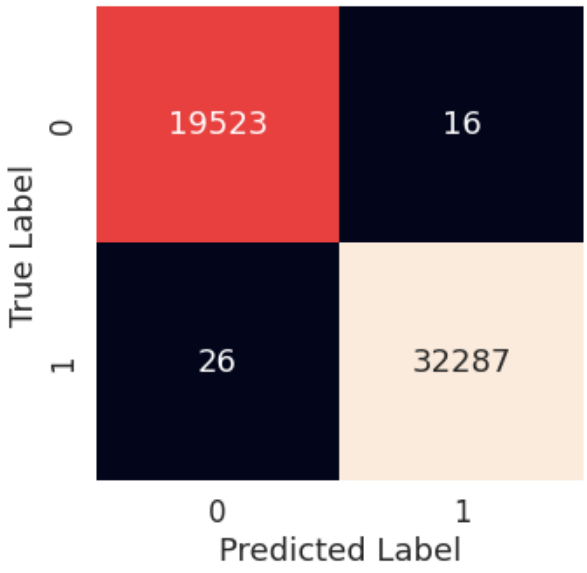

Figure 13: Confusion matrix of RF classifier

Figure 13 shows the prediction provided by the RF model. The anticipated outcome and the model's calculated performance are shown in the confusion matrix. 51810 predictions were accurate, while 42 were wrong.

6.1.4 Adaboost

The classification accuracy of the Adaboost algorithm has been shown in figure 14.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19539
1	1.00	1.00	1.00	32313
accuracy			1.00	51852
macro avg	1.00	1.00	1.00	51852
weighted avg	1.00	1.00	1.00	51852

Figure 14: Classification accuracy of Adaboost classifier

The accuracy of the AB algorithm was 99.34 percent. For both normal and intrusion data prediction, the F1 score is 100 percent. For both 0 and 1, precision is 100 percent. Also, for both 0 and 1, recall is 100 percent. The AB algorithm's confusion matrix is shown in Figure 15.

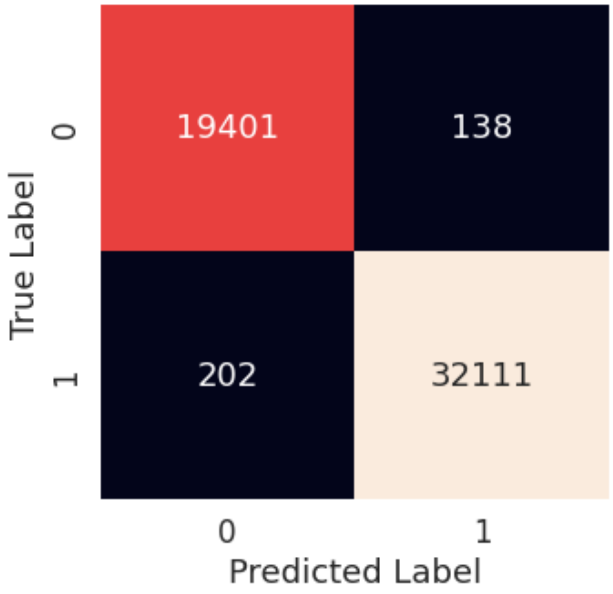

Figure 15: Confusion matrix of Adaboost classifier

Figure 15 shows the prediction given by the AB model. The anticipated outcome and the model's calculated performance are shown in the confusion matrix. There were 51512 accurate and 340 erroneous guesses.

6.2 Model Comparison

The models are compared to those previously researched, as shown in Table 1. The Decision tree and Random forest classifiers beat all others in the system, as seen in the table.

Table 1: Performance Comparison

This Paper (Model Name)	Accuracy (%)	Reference Paper (Model Name)	Accuracy (%)
Decision Tree	**99.87**	Ref [22] Neural Fuzzy	95.3
Adaboost	99.34	Ref [23] MLP	94.5
Gradient Boosting Classifier	99.72	Ref [24] Naïve Bayes	89
Random Forest	99.92	Ref [25] K-Nearest Neighbor	78

Summary:

Despite the fact that all of the methods in Table 1 are accurate, it is clear that only the decision tree and random forest classifier approaches provide a considerable benefit in terms of precision. Using a decision tree and random forest classifier, this work achieved 99.87 and 99.92 percent accuracy, respectively. However, the accuracy of NB and MLP ref [24] and [23] was 89 percent and 94.5 percent, respectively. Using KNN, this work achieved 99.56 percent accuracy, whereas ref [25] achieved 78 percent accuracy using the same model.

Chapter 7

conclusion

7. Conclusion

At this moment, network security is a major concern for all businesses and institutions. Despite the development of multiple methods to ensure the infiltration of intrusions to the network infrastructure, intrusions continue to attempt to gain successful access to many companies' data networks and web services. Intrusion detection systems (IDS) are designed to make dealing with attacks easier for computer systems. The evolution and expansion of machine models enable the facility to detect the intrusion early and prevent the dire consequences that happen due to intrusion and cyber-attacks. This study focuses on the efficiency of several ML algorithms in appropriately anticipating intrusions caused by a number of numerical variables. By using Random Forest, Decision Tree, Gradient boosting, and Adaboost classifier this research achieved 99.92, 99.87, 99.72, and 99.34 percent accuracy, respectively. The framework models may be boosted in the future using a larger dataset and machine learning and deep learning models such as ExtraTreesClassifier, Voting Classifier, MLP, LSTM, and others. This will improve the framework's constancy as well as its demonstration. The future of intrusion detection systems will be to interpret and apply these techniques for detecting new and emerging threats. The general public may have the assistance of a machine learning architecture to evaluate the probability of a network intrusion transpiring in exchange for just offering some basic info. In an ideal world, it would assist businesses in obtaining early detection of intrusions, allowing them to protect highly classified information and financial resources.

BIBLIOGRAPHY

References

[1] D. E. Denning, “An intrusion-detection model,” IEEE Transactions on software engineering, no. 2, pp. 222–232, 1987.

[2] B. A. Tama and K.-H. Rhee, “Attack classification analysis of iot network via deep learning approach,” Research Briefs on Information Communication Technology Evolution (ReBICTE), vol. 3, no. 15, pp. 1–9, 2017.

[3] “What is an IDS? Intrusion Detection Systems Explained | AT&T Cybersecurity,’’ cybersecurity.att.com, [Online]. Available: https://cybersecurity.att.com/solutions/intrusion-detection-system/ids-explained.

[4] “NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB,” University of New Brunswick, [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html.

[5] S. D. Bay, D. F. Kibler, M. J. Pazzani, and P. Smyth, “UCI Machine Learning Repository: KDD Cup 1999 Data Data Set,”, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data.

[6] L. Mehra and M. K. Gupta, “An effectual & secure approach for the detection and efficient searching of Network Intrusion Detection System (NIDS),” in *Proc. 2015 International Conference on Computer, Communication and Control (IC4)*, Indore, India, pp. 1-5, 2015.

[7] H. Liu and B. Lang, “Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,” *Applied Sciences*, vol. 9, no. 20, pp. 4396, 2019.

[8] H. Gharaee and H. Hosseinvand, “A new feature selection IDS based on genetic algorithm and SVM,” in *Proc. 2016 8th International Symposium on Telecommunications (IST)*, Tehran, Iran, pp. 139-144, 2016.

[9] Y. Ping, H. Yafei, Z. Yiping, Z. Shiyong and D. Zhoulin, “Flooding attack and defence in ad hoc networks,” *Journal of Systems Engineering and Electronics*, vol. 17, no. 2, pp. 410-416, 2006.

[10] P. Yi, T. Zhu, Q. Zhang, Y. Wu and J. Li, "A denial of service attack in advanced metering infrastructure network," in *Proc. 2014 IEEE International Conference on Communications (ICC),* Sydney, NSW, Australia, pp. 1029-1034, 2014

[11] P. Yi, Z. Dai, S. Zhang and Y. Zhong, "A New Routing Attack in Mobile Ad Hoc Networks," *International Journal of Information Technology,* vol. 11, no. 2, pp. 83-94, 2017.

[12] P. Yi, F. Zou, V. Zou and Z. Wang, "Performance analysis of mobile ad hoc networks under flooding attacks," *Journal of Systems Engineering and Electronics,* vol. 22, no. 2, pp. 334-339, 2011.

[13] P. Yi, Y.-k. Zhou, Y. Wu and N. Liu, "Effects of denial of service attack in mobile ad hoc networks," *Journal of Shanghai Jiaotong University (Science),* vol. 14, no. 5, pp. 580-583, 2009.

[14] P. Yi, J. Cai, Y. Wu and Y. Li, "Impact of two kinds of DoS attacks on mobile ad hoc networks," *Journal of Computational Information Systems,* vol. 5, no. 5, pp.1433-1443, 2009.

[15] P. Yi, T. Zhu, N. Liu, Y. Wu and J. Li, "Cross-layer Detection for Black Hole Attack in Wireless Network," *Journal of Computational Information Systems,* vol. 8, no. 10, pp. 4101-4109, 2012.

[16] P. Yi, Z. Wang and Y. Wu, "Intrusion Detection for Wireless Mesh Networks using Finite State Machine," *China Communication,* vol. 5, no. 5, pp. 40-48, 2010.

[17] Y. Ping, J. Xinghao, W. Yue and L. Ning, "Distributed intrusion detection for mobile ad hoc networks," *Journal of Systems Engineering and Electronics,* vol. 19, no. 4, p. 851–859, 2008.

[18] P. Shanghai, J. Tong, T. Zhu, Y. Wu, P. Yi *et al.*, "An Intrusion Prevention Mechanism in Mobile Ad Hoc Networks," *Ad Hoc & Sensor Wireless Networks,* vol. 17, no. 3, pp. 269-292, 2013.

[19] Y. Ping, Z. Futai, J. Xinghao and L. Jianhua, “Multi-agent cooperative intrusion response in mobile adhoc networks,” *Journal of Systems Engineering and Electronics*, vol. 18, no. 4, p. 785–794, 2007.

[20] P. Yi, T. Zhu, Q. Zhang, Y. Wu, and J. Li, “Green firewall: An energy-efficient intrusion prevention mechanism in wireless sensor network,” in *Proc. 2012 IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, USA, pp. 3037–3042, 2012.

[21] LUFlow Intrusion detection dataset. Available on: https://www.kaggle.com/datasets/mryanm/luflow-network-intrusion-detection-data-set

[22] A. N. Toosi and M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers," 2007. [Online]. Available: https://www.researchgate.net/publication/222533954_A_new_approach_to_intrusion_detection_based_on_an_evolutionary_soft_computing_model_using_neuro-fuzzy_classifiers. [Accessed 22 11 2021].

[23] U. Serpil, T. Zeynep and A. A. Muhammed, "Intrusion Detection System with Recursive Feature Elimination by Using Random Forest and Deep Learning Classifier," 24 01 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8625318. [Accessed 20 11 2021].

[24] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," Artificial intelligence and statistics, p. 1273–1282, 2017

[25] A. Ajayi, "Comparative study of selected data mining algorithms used for intrusion detection," 06 2013. [Online]. Available: https://www.researchgate.net/publication/331304899_Comparative_study_of_selected_data_mining_algorithms_used_for_intrusion_detection. [Accessed 19 11 2021].