**Department of Electrical and Computer Engineering**

**North South University**



**Senior Design Project**

**Software Defined Networking (SDN) Intrusion Detection Using Machine Learning Models**

|  |  |
| --- | --- |
| **Muhammad Zawad Mahmud**  **Samiha Islam** | **ID # 1931401042**  **ID # 1931393042** |
| **Md. Solayman Hossain** | **ID # 1931565042** |

**Faculty Advisor**

**Dr. Mohammad Monirujjaman Khan**

**Associate Professor**

**Department of Electrical and Computer Engineering**

**North South University**

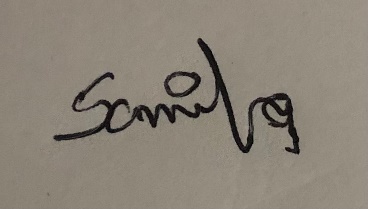
**Fall 2022**

**DECLARATION**

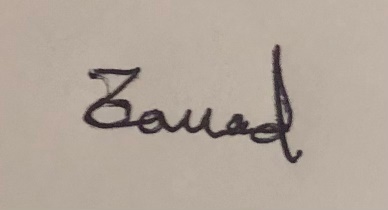
This is to certify that this Project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

**Students’ Name & Signature**

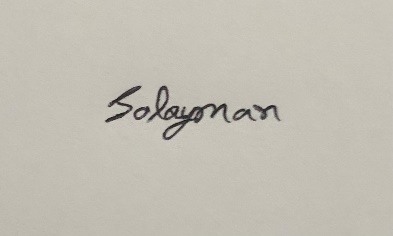
1. **Samiha Islam**



1. **Muhammad Zawad Mahmud**



1. **Md. Solayman Hossain**



**APPROVAL**

The direct research project entitled “**Software Defined Networking (SDN) Intrusion Detection Using Machine Learning Models**” by **Muhammad Zawad Mahmud (ID#1931401042), Samiha Islam (ID#1931393642)** and **Md. Solayman Hossain (ID#1931565042)** is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on January 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’s Signature** |
| **Dr. Rajesh Palit**  **Professor & Chair**  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 senior design 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 CSE499 in which we could really work on this research 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**

Information security and data analysis systems for Big Data have recently taken on new significance because of the massive amount of data and its steady growth. The volume, velocity, and variety of Big Data necessitate the development of novel methods for dealing with it. An intrusion detection system (IDS) is a piece of hardware or software that looks at data and looks for any attack on a system or network. SDN anomaly detection method is a software program that uses machine algorithms to identify manufacturing anomalies. Anomaly detection is used to solve a variety of problems, including selective logging, privacy protection, reputation-based protection, multiple threat protection, and dynamic threat response. Distributed networks that create huge volumes of data on a daily basis include mobile phones, wearable gadgets, and self-driving automobiles. Anomaly detection services are critical for the device's security and privacy. Machine learning is a subset of artificial intelligence that allows software programs to anticipate outcomes more accurately without having to build them explicitly. Machine learning with network anomaly detection systems has gotten a lot of attention because of its high categorization accuracy. As a consequence of technological advancements, cyberattacks are expected to rise considerably in frequency by 2021. This research looks at a water manufacturing anomaly detection system that employs machine learning to improve its efficiency and accuracy. SDN dataset has been used in this research. By analyzing the combinations of the most popular feature selection techniques and classifiers, such as K-Nearest Neighbors (KNN) Classification, Decision Tree (DT) Classification, and Random Forest (RF) Classification, it has a good union of feature selection techniques and classifiers. The decision tree was found to be 98 percent accurate, while K-Nearest Neighbors (KNN) was 99 percent accurate, Random Forest was 95 percent accurate. The experiment demonstrated that the machine learning model is effective for big data, has high performance, and requires less time to train.

**Table of Content**

|  |  |  |
| --- | --- | --- |
|  |  | Page |
|  | **Chapter Introduction**…………………………………………………. | 11 |
| 1.1 | Introduction………………………………………………………………………… | 12 |
| 1.2 | Project Details……………………………………………………………………… | 14 |
| 1.3 | Project Goals……………………………………………………………………… | 17 |
| 1.4 | Summary…………………………………………………………………………… | 17 |
|  |  |  |
|  | **Chapter 2: Motivation**………………………………………………. | 18 |
| 2.1 | Introduction………………………………………………………………………… | 19 |
| 2.2 | Motivation towards our project……………………………………………………. | 19 |
| 2.3 | Summary…………………………………………………………………………… | 20 |
|  |  |  |
|  | **Chapter 3: Related Works**…………………………………………... | 21 |
| 3.1 | Introduction………………………………………………………………………… | 21 |
| 3.2 | Systems related to our project……………………………………………………… | 21 |
| 3.3 | Problems with the current system…………………………………………………:,., | 22 |
| 3.4 | Proposed Solution………………………………………………………………...... | 22 |
| 3.5 | Summary………………………………………………………………………….. | 22 |
|  |  |  |
|  | **Chapter 4: Methodology** ………………………………………... | 23 |
| 4.1 | Introduction………………………………………………………………………… | 24 |
| 4.2 | Technical Design: System Level…………………………………………………… | 24 |
| 4.3 | Summary…………………………………………………………………………… | 26 |
|  |  |  |
|  | **Chapter 5: Algorithm** **analysis** …………………………………………. | 27 |
| 5.1 | Introduction……………………………………………………………… | 28 |
| 5.2 | Algorithm Structural……………………………………………………… | 28 |
| 5.3 | Summary…………………………………………………………………………… | 30 |
|  |  |  |
|  | **Chapter 6: Results and data analysis** … ……… | 34 |
| 6.1 | Introduction……………………………………………………………… | 35 |
| 6.2 | Result analysis and model comparison………………………………………… | 40 |
| 6.3 | Summary………………………………………………………………………….. | 45 |
|  |  |  |
|  | **Chapter 7: Skills**……………………………………………………... | 47 |
| 7.1 | Introduction……………………………………………………………… | 47 |
| 7.2 | Skills obtained……………………………………………………………………... | 48 |
| 7.3 | Summary………………………………………………………………… | 49 |
|  | **Chapter 8: Future Work**……………………………………………. | 50 |
| 8.1 | Introduction………………………………………………………………................ | 50 |
| 8.2 | Future scope of work………………………………………………………….......... | 51 |
| 8.3 | Summary…………………………………………………………............................ | 51 |
|  |  |  |
|  | **Chapter 9: Design Impact**………………………………………… | 52 |
| 9.1 | Introduction………………………………………………………………………… | 53 |
| 9.2 | Environmental Impact……………………………………………………………… | 53 |
| 9.3 | Economic Impact…………………………………………………………………… | 53 |
| 9.4 | Social Impact……………………………………………………………………….. | 53 |
| 9.5 | Sustainability……………………………………………………………………… | 53 |
| 9.6 | Summary…………………………………………………………………………… | 54 |
|  |  |  |
|  |
|  |  |  |
|  | **Chapter10: Compliance with standard**… … ……… ………… | 55 |
| 13.1 | Introduction………………………………………………………………………… | 55 |
| 13.2 | Compliance with IEE standard……………………………………………………. | 55 |
| 13.3 | Summary………………………………………………………………………….. | 57 |
|  | **Chapter 11: Conclusion**……………………………………………… | 58 |
|  |  |  |
|  | **Bibliography**…………………………………………………………... | 60 |
|  |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Figure No. | Figure caption | Page No. |
|  |  |  |
| 1 | Block Diagram of the System | 22 |
| 2 | Model of Decision Tree Classifier | 26 |
| 3 | Model of Random Forest Classifier | 28 |
| 4 | Model of Gradient Boosting Classifier | 29 |
| 5 | Model of Ada Boosting Classifier | 32 |
| 6 | Classification accuracy of Gradient boosting | 36 |
| 7 | Confusion Matrix of Gradient boosting | 36 |
| 8 | Confusion Matrix of Decision Tree Classifier | 38 |
| 9 | Confusion Matrix of Random Forest | 37 |
| 10 | Classification accuracy of Random Forest | 37 |
| 11 | Classification accuracy of Ada Boosting | 39 |
| 12 | Classification accuracy of Decision Tree | 38 |
| 13 | Confusion Matrix of Ada Boosting | 39 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| 1 | Comparison Table | 40 |

**CHAPTER 1**

**Introduction**

**Introduction**

Detecting anomalies can be difficult, especially with labeled datasets, because there is some human bias involved in labeling the final product as anomalous or good. These massive production systems must be monitored every 10 milliseconds in order to capture their activity, which results in a flood of data and what we term the Industrial Internet of Things (IIOT). Furthermore, no manufacturer wants to make an outlier product. As a result, the anomalies are like a needle in a haystack, resulting in a substantially unbalanced dataset. Because of the rapid advancement of technologies such as the Internet, Internet-of-Things (IoT), and communication systems, hackers have evolved at a faster rate in terms of their skills. These crooks are always looking for new methods to breach the security of computer networks. As a result, intrusion detection systems (IDSs) have become inextricably linked to the security of a computer network. An IDS is a hardware or software system that monitors a company's computer network for threats or assaults, both real and prospective. Furthermore, an IDS is capable of reacting to and reporting any fraudulent transactions [1]. Node or host-based IDS (HIDS), network or distributed IDS (DIDS or NIDS), and hybrid IDS are the three major categories of IDS (HYIDS). This categorization is based on the operating philosophy of the IDS. HIDS is a security system that works on a single node computer system and is concerned with the security of its host. A DIDS, on the other hand, operates on a dispersed computer network and analyzes network traffic for suspicious activity. DIDS and HIDS are combined in HYIDS [2-3]. In addition, signature-based, anomaly-based, and hybrid-based IDSs may also be divided into three groups. In order to identify intrusions, a signature-based IDS employs an existing data store of previously intercepted assaults. An anomaly-based IDS, on the other hand, is concerned with network activity. It is constantly on the lookout for unusual and fag-like behavior. Finally, a hybrid-based IDS combines anomaly-based and signature-based IDSs [4–6]. An IDS's key philosophical design aim is to reduce the number of false-positive alerts while increasing detection accuracy. As a result, that concept must be considered while designing and implementing an IDS [7]. In recent years, machine learning (ML) based intrusion detection systems have emerged as the top systems in the intrusion detection research arena. Machine learning (ML) allows computers to learn and improve based on past data. To put it another way, ML-based computer applications do not need explicit engineering (programming). They have the ability to learn on their own [8]. In general, there are two types of machine learning philosophies: supervised and unsupervised machine learning. In supervised machine learning [9], models are trained using labeled data. The data used to train models in unsupervised machine learning is unstructured [10]. This study’s goal is to look at supervised machine learning approaches, especially for binary and multiclass classification problems. When a supervised ML model is asked to predict a discrete value, the classification operation occurs [11]. In this configuration, the datasets used to train the models are often huge and have a high-dimensional feature space. Because of this complexity, training and testing supervised machine learning models may take a long time. As a result, feature engineering methods are crucial for reducing the number of features necessary for the training and testing stages [2, 12]. The main contribution of this paper is that we used the supervised ML techniques for IDS, k-Nearest-Neighbor (kNN), Logistic Regression (LR), Random Forest, Support Vector Machine (SVM), and Decision Tree in our experiments (DT). And the novelty of this research is that we use all ML methods to figure out how important each feature is in the UNSW-NB15 dataset [13, 14]. This way, we can build a smaller and more efficient feature vector.

.

**1.1 Supervised Learning**

The model is trained using a labeled dataset. It has input data as well as output data. The data is categorized and divided into training and test datasets. The training dataset is used to train our model, while the testing dataset is used to test the model's accuracy. There is a dataset comprising models and their output. Its examples include categorization and regression.

**1.2 Unsupervised Learning**

The dataset does not include any classification or labeling of the data used to train. The goal is to uncover hidden patterns within the data. The model has been conditioned to recognize patterns. It can readily anticipate hidden patterns for each new input dataset, but when it comes to describing hidden patterns, it draws conclusions from datasets. There are no responses in the dataset while using this strategy. Unsupervised learning techniques, such as clustering, provide an example.

**1.3 Reinforcement Learning**

It doesn't use a labelled dataset, and the outcomes aren't tied to data, so the model learns from its mistakes. In this method, the model improves its presentation based on its relationship to the environment and determines how to address its flaws and achieve the best result by assessing and testing numerous options.

**1.4 Classification Machine Learning Techniques**

The classification task is used to make predictions about future cases based on previous data. Researchers have used a variety of data mining approaches such as Nave Bayes, neural networks, and decision trees to achieve a precision diagnosis in heart disease. The accuracy provided by various methodologies varies depending on a variety of factors. This study provides a diagnostic accuracy score that can be used to enhance health outcomes. The classifiers in this study were compared with each other for better interpretation.

Machine learning techniques have been around us and has been compared and used for analysis for many kinds of data science applications. The major motivation behind this research-based project was to explore the feature selection methods, data preparation and processing behind the training models in the machine learning. With first hand models and libraries, the challenge we face today is data where beside their abundance, and our cooked models, the accuracy we see during training, testing and actual validation has a higher variance. Furthermore, as the whole machine learning is motivated to develop an appropriate computer-based system and decision support that can aid to early detection of heart disease, in this project we have developed a model which predicts if patient will have heart disease based on various features (i.e., potential risk factors that can cause heart disease) using various classifiers like Decision trees, Random Forest, K-nearest neighbor and Logistic regression. Hence, the early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine.

The main objectives of developing this study are:

• To develop machine learning model to predict future possibility of heart disease by implementing various machine learning algorithms and comparing them.

• To determine significant risk factors based on medical dataset which may lead to heart disease.

• To analyze feature selection methods and understand their working principle.

**CHAPTER 2**

**Motivation**

**2.1 Introduction**

In this chapter we discuss the motivation due to which we thought of implementing this system. We will also discuss in this chapter as to why we have chosen the medical field apart from all other fields to work it.

**2.2 Motivation towards our project**

Machine learning techniques have been around us and has been compared and used for analysis for many kinds of data science applications. The major motivation behind this research-based project was to explore the feature selection methods, data preparation and processing behind the training models in the machine learning. With first hand models and libraries, the challenge we face today is data where beside their abundance, and our cooked models, the accuracy we see during training, testing and actual validation has a higher variance. Furthermore, as the whole machine learning is motivated to develop an appropriate computer-based system and decision support that can aid to early detection of intrusion detection, in this project we have developed a model which predicts if intrusion is taking place based on various features using various classifiers like Decision trees, Random Forest, Gradient Boosting and Ada Boosting. Hence, the early prognosis of intrusion can aid in making decisions in order to save many data lose.

**2.3 Summary**

This chapter provided the idea about the motivation towards our project which aims to predict intrusion in software defined networking.

**CHAPTER 3**

**Related Works**

**3.1 Introduction**

In this chapter we discuss the types of intrusion detection systems that currently exist in the market. We also focus on the loopholes that the current system entails and a proper justification will be provided as to why our system is the ideal one in the current circumstances.

**3.2 Related Work to our project**

Over the UNSW-N15 and KDDCup99 datasets, the researchers used the Genetic Algorithm (GA) in combination with the Logistic Regression (LR) internally developed feature selection approach [15]. The Weka simulation tool was employed in this study. The GA-LR combined with the DT classifier achieved a detection score of 81.42 percent and a FAR of 6.39 percent using 20 characteristics out of the 42 features included in the UNSW-NB15 feature space after numerous simulations. As part of a team, the GA-LR and the DT classifier detected 99.90% of the KDDCup99 dataset with 99.990% accuracy, and 0.105% of the dataset had a false alarm rate (FAR). The authors of [16] described a filter-based strategy for detecting Distributed Denial of Service (DDoS) that used numerous filters. Information Gain, Chi-Square and Gain Ratio, and ReliefF were among the filter techniques employed. The NSL-KDD attack detection dataset was used by the researchers to illustrate the system's functionality. The authors used the Decision Tree (DT) algorithm for classification, which was trained and verified using the k-fold cross-validation approach with k = 10. The testing findings indicated that the DT classifier was able to achieve a detection accuracy score of 99.67 percent and a false alarm rate (FAR) of 0.42 percent using just 13 features out of 42 features (full feature space). However, this work did not go into great detail on the NSL-multiclass KDD's classification challenge. An IDS was combined with a filter-inspired input reduction technique in [17]. Different datasets were used in this study, including the Kyoto 2006, the KDDCup99, and the NSL-KDD. In this study, the authors used a Flexible Mutual Information (FMI) approach to proving the existence of a link between multiple input variables. A non-linear correlation metric is the FMI. The Least Square SVM classifier was utilized in the trials (LS-SVM). According to the data, the LS-SVM FMI produced a FAR of 0.28 percent and an accuracy of 99.94 percent for the NSL-KDD with 18 features. In iteration 10, the LS-SVM FMI achieved an overall accuracy of 78.86 percent in the case of KDD Cup 99 and a detection rate of 97.80 percent with a FAR rate of 0.43 percent in the case of Kyoto 2006 +. The authors of [18] used a filter-based technique methodology to develop an IDS with the goal of reducing the number of input characteristics (features) necessary for training and testing their model. A correlation input selection strategy was employed in conjunction with the DT classifier. The NSL-KDD dataset was used in the experimental procedures. After applying the filter to the feature space, 14 features were chosen. In addition, the author investigated the binary classification configuration as well as the multiclass classification setup, which covered all five kinds of assaults inside the NSL-KDD. An accuracy rate of 83.66 percent was achieved in the multiclass configuration and 90.30 percent in the binary setting. Using the Pigeon Inspired Optimizer, the researchers built a feature reduction approach for intrusion detection systems in [19]. (PIO). The PIO is a bio-inspired algorithm inspired by white pigeon fights. These birds continually refer to the best bird in the flock while deciding where to fight [20]. The following two forms of PIO were evaluated in this study: Sigmoid PIO and Cosine PIO. The intrusion detection sets UNSW-NB15, NSL-KDD, and KDDCup99 were examined. The Sigmoid PIO chose ten characteristics from the KDDCup99, whereas the Cosine PIO chose seven. The NSL-KDD was divided into 18 features by the Sigmoid PIO and 5 features by the Cosine PIO. The UNSW-NB15 was divided into 14 features by the Sigmoid PIO and 5 features by the Cosine PIO. Using the KDDCup99, the Sigmoid PIO achieved 94.7 percent accuracy, the NSL-KDD achieved 86.9% accuracy, and the UNSW-NB15 achieved 91.3 percent accuracy. Cosine PIO, on the other hand, received a score of 96.0 percent on the KDDCup99, 88.3% on the NSL-KDD, and 91.7 percent on the UNSW-NB15. Janarthanan and Zargari [21] used the UNSW-NB15 to construct a variety of feature selection algorithms in order to find the best feature space. The attribute evaluator, greedy stepwise, and information gain algorithms, as well as the Ranker Method, were built using the Weka tool. Two subsets were selected after several simulations. To assess the efficacy of each subgroup, the authors employed the Kappa statistic. During the studies, a variety of classifiers were investigated; nevertheless, the RF classifier was chosen as the best approach in terms of overall performance. Over the test dataset, the initial subset with eight significant characteristics has a Kappa Score of 0.6891 and an efficiency of 75.6617 percent. With just five important characteristics, the second group got a Kappa value of 0.7639 and an efficiency of 81.6175 percent. Vikash and Ditipriya [22] developed an IDS system that they tested on the UNSW-NB15 dataset. The authors examined a feature reduction strategy influenced by the Information Gain methodology in this study. The 22 most essential qualities were chosen using a filter-based feature extraction approach. Furthermore, the IDS suggested in this paper used an integrated rule-based approach that used numerous Tree-based classifiers to complete the classification process. The Attack was used to evaluate the system's performance. The test data accuracy, the F-Measure (FM), and the False Alarm Rate (FAR). The suggested IDS has an accuracy of 57.01 percent, an FM of 90 percent, and a FAR of 2.01 percent, according to the data. When evaluating other ML algorithms that may be included into a substitute for strict adherence to Tree-based approaches, the findings achieved in this study can be improved. Particle Swarm Optimization (PSO), Firefly Optimization (FO), Grey Wolf Optimization (GO), and Genetic Technique were used to create a feature extraction algorithm in [23]. These approaches were applied to the UNSWNB15 dataset repeatedly in order to determine which feature subset would give the best attack detection accuracy. A feature subset of 30 characteristics was chosen after multiple attempts. In addition, the classification procedure was carried out using the J48 tree-based model and SVM models. Accuracy, false-positive rate (FPR), false-negative rate (FNR), and false-positive rate (FM) were the key performance parameters addressed in this research. The binary classification system was used in the studies, which were conducted on the UNSW-NB15 training subset. The suggested J48 model has a training accuracy of 90.484 percent, an FPR of 14.950 percent, and an FM of 90.172 percent, according to the data. The suggested SVM model also has a training accuracy of 90.119 percent, an FM of 89.808%, an FPR of 15.391 percent, and an FNR of 3.130 percent. In this investigation, we utilized five machine learning algorithms, and all of them were accurate to better than 94 percent. With the decision tree classifier and support vector classifier, we achieved the highest accuracy of 100 percent. As a consequence, we were able to reduce the accuracy gap in our study. The use of these five models has the benefit of allowing for comparative examination. These comparisons help us figure out which model has the highest degree of accuracy. Any company may use our technology to identify intrusions by simply implementing them. It will save a lot of personal information, privacy, and money for any firm. The rest of the article is as follows: Section 2 describes the experimental approach and procedures, whereas Section 3 analyzes the results and Section 4 discusses the conclusion.

**CHAPTER 4**

**Methodology**

**4.1 Introduction**

In this chapter we discuss the aspect of the technical design of our system. By going through the system level design it would be easier to conceptualize the entire data flow of the system.

**4.2 Technical Design: System Level**

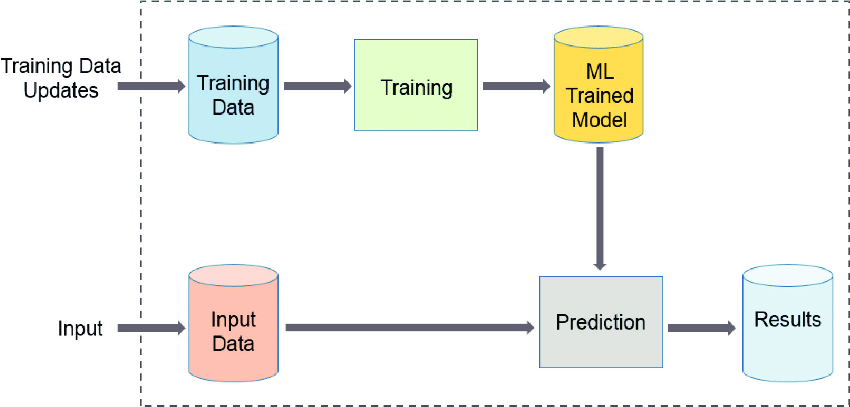


Figure 1

Figure 1 shows a typical ML workflow in which the model is trained on training data before being used to forecast results from new data. Based on the kind of data labeling, ML techniques have been comprehensively separated into supervised (such as classification or regression), unsupervised (such as clustering and probability density function estimation), and semi-supervised learning approaches. In the age of big data, the application of machine learning techniques in radiation oncology research is growing quickly. Applications include organ division, picture direction, movement following, quality affirmation, treatment reaction exhibiting, and more. An ML algorithm is a computer procedure that performs the intended goal without being explicitly programmed by using input data (also known as "hard-coded"). These algorithms are "soft-coded" in a way since they automatically modify or alter their design as a result of repeated use to become more effective at the task at hand. The process of adaptation known as training involves providing input data samples and desired results. The algorithm then fine-tunes its settings to make sure that in addition to producing the intended result when given the training data, it can generalize to do so when working with new data.

**4.3 Summary**

As mentioned earlier, the technical design has enabled us to get a clear picture of how our system is operating. Therefore considering the above data flow diagram we can comprehend the method in which our system is being operated.

**CHAPTER 5**

**Algorithm**

**5.1 Introduction**

In this chapter we discuss the algorithm of our system.

**5.2 Approach**

In this chapter, we discuss our heart disease prediction model architecture in detail. We show the outcomes of our different experiments as well.

**5.1** **Decision Tree**

Decision Tree, Random Forest, and Logistic Regression are three machine learning methods that are used to make this prediction. When the user enters all of the symptoms, he must press the buttons for the appropriate algorithm; for example, if the user enters all of the symptoms and only presses the Random Forest button, the result will be calculated using that algorithm; similarly, we have used three algorithms to provide a clearer picture of the results, and the user must be satisfied with his predicted disease result.

Decision tree is the supervised learning algorithm and it uses class-labelled training tuples/rows.

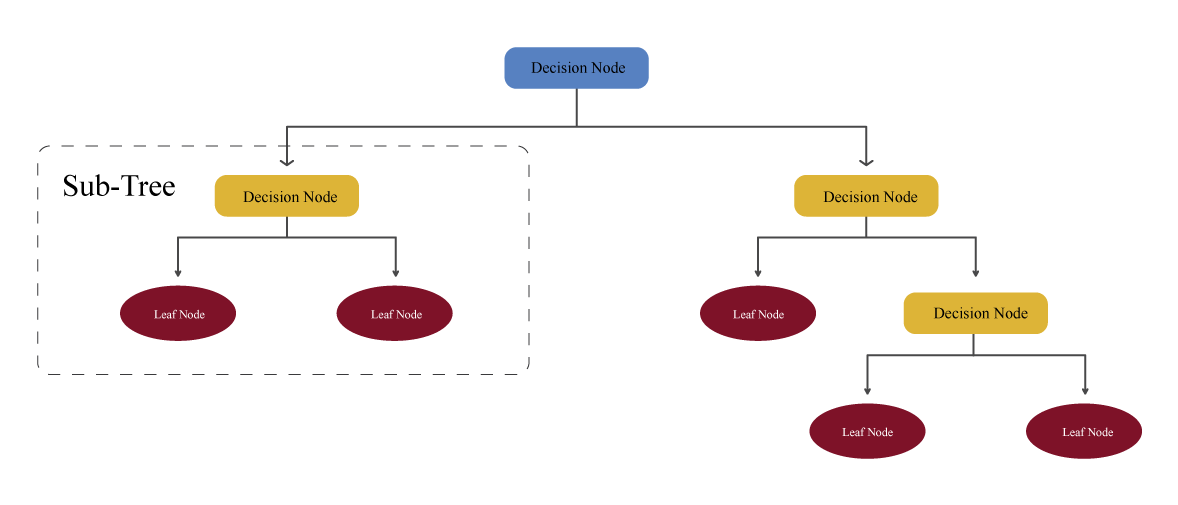


Figure 2: Model of Decision Tree Classifier

A decision tree algorithm is a flowchart-like above tree structure and useful for both regression and classification. First, the root/node is prepended to the entire training set. Categorical values are expected for attribute values. If the values are continuous, they must be converted to discrete before the model can be created. Recursively, records and rows are assigned depending on attribute/function values. To identify where to divide a node into two or more sub-nodes, a decision tree uses numerous procedures.

The homogeneity of the ensuing sub-nodes is increased/increased as sub-nodes are formed. In other words, the node's purity rises in proportion to the output variable. The decision tree separates the node into all accessible variables before selecting the sub-node with the most homogeneous sub-nodes. The following criteria can be used to choose attributes: Entropy, Information Gain, Gini Index, Gain Ratio, and Chi-Square Reduction in Variance.

**5.2 Random Forest**

Random Forest, a supervised learning technique that can be used for both regression and classification, was the next algorithm we used. The training dataset is randomly partitioned for building decision trees. Random subsets of characteristics are taken when separating the nodes. It is referred regarded as "random" because of these two variables. The classifier is a random forest, which is made up of a collection of tree-structured classifiers, identically distributed random trees, with each random tree containing a unit of vote/poll for input categorization. Each tree makes use of each other, and to split the tree and establish the final label, either entropy or the Gini index is utilized. The final output of each tree is combined and voted on by the weighted values to create the final classifier with the highest accuracy.

The Random Forest algorithm works as follows: A random data set is chosen, which distributes a collection of samples from the training dataset at random while maintaining the class distribution.

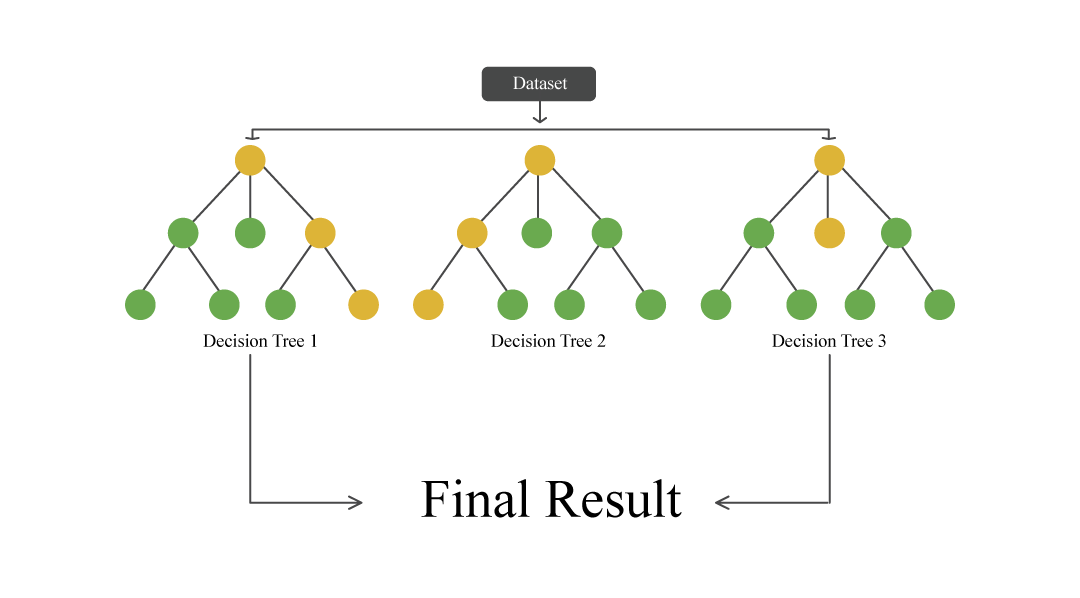


Figure 3: Model of Random Forest Classifier

Logistic Regression is a supervised machine learning algorithm that predicts using a logit/sigmoid function. The function returns values between 0 and 1, and logistic regression employs a threshold value to determine if the output is one or zero. If the sigmoid function's output is greater than the threshold value, our output is one; otherwise, it is zero. Log loss is the loss function used in logistic regression. Although the name implies that it is a regression, it was originally designed to solve a classification problem. It's particularly well suited to binary classification.

**5.3 Gradient Boosting**

The gradient boosting approach [31], abbreviated as GBA, is one of the most successful machine learning methods. 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 adapts a new model to the regression produced by the preceding predictor [32]. Fig. 5 shows the GB algorithm's schematic design.

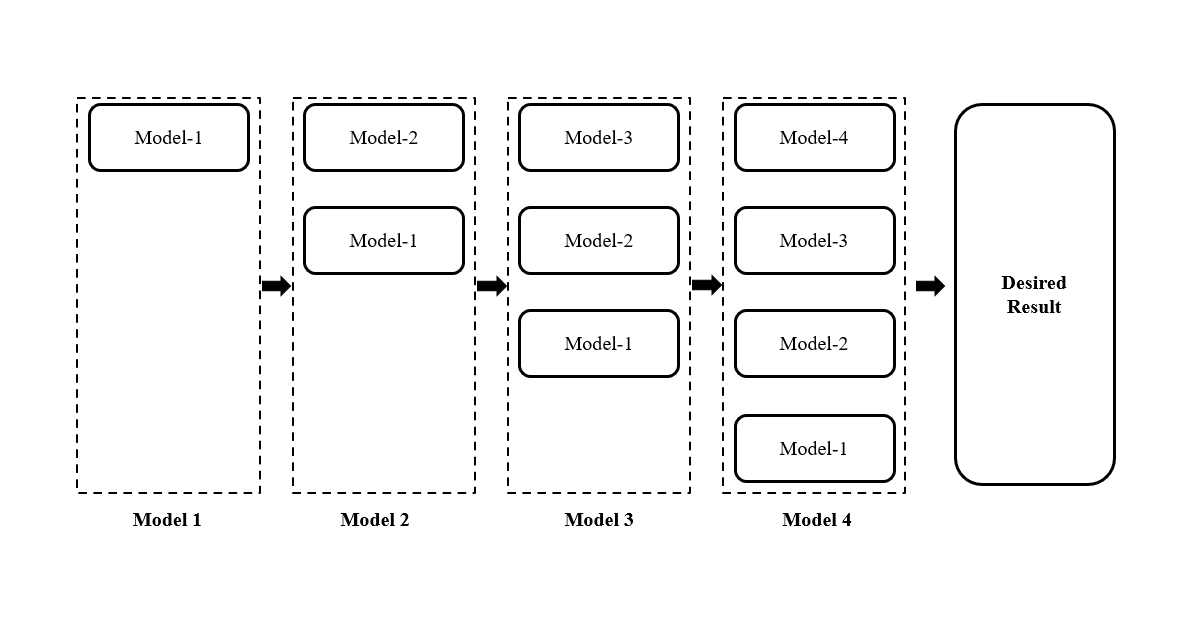


Figure 4: Diagram of Gradient Boosting Classifier

Preprocessing of data is not a prerequisite in GBM. 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 Ada Boosting**

The Ada Boosting algorithm [33] is a boosting approach that is used in machine learning as an ensemble method. As weights are reassigned to each instance in this process, with heavier weights applied to erroneously identified instances [34]. The subsequent learners, with the exception of the first, are developed from the prior ones. Simply put, poor learners develop into strong learners.

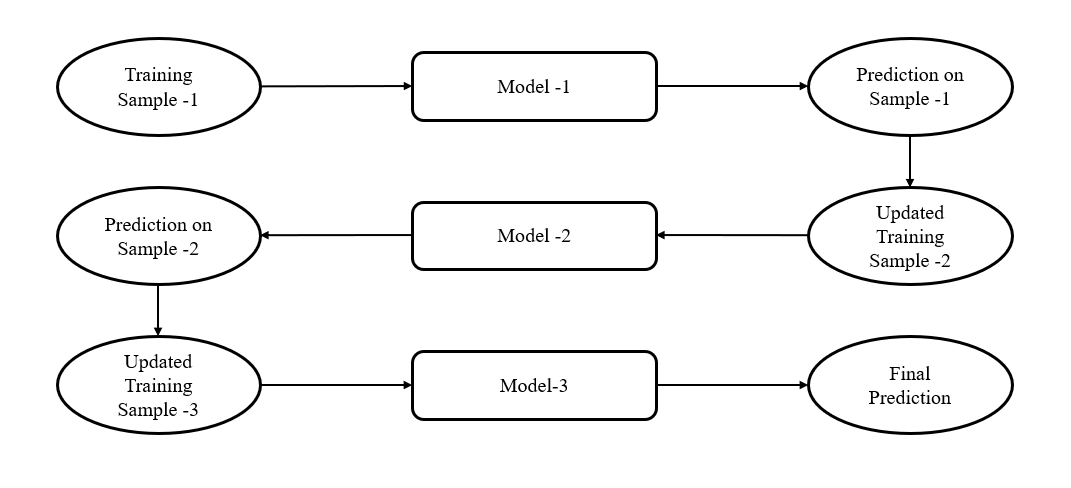


Figure 6: Diagram of Ada Boosting Classifier

Fig. 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 Fig. 7, three models are created by including the mistakes from the preceding model. This is the mechanism through which boosting operates.

**CHAPTER 6**

**Result and Data Analysis**

**6.1 Introduction**

In this chapter we explore the experimental result. This study intends to assess the likelihood of developing intrusion detection. The use of several machine learning algorithms on the data set and dataset analysis are explored in this research study to achieve the goal. This research also shows which characteristics contribute more than others to the prediction of higher precision. This could save money by avoiding the cost of multiple trials for a patient, as all of the qualities may not play a significant role in predicting the outcome.

The correlation heatmap of the attributes was plotted before selecting the features to be employed in the prediction model. Correlation is a statistical measure of a linear relationship between two variables. It's also known as the dependency measure between two variables. The correlation coefficient, often known as the correlation coefficient, is a measurement of how two different variables' values change in relation to one another.

**Model Accuracy**

**Gradient Boosting**

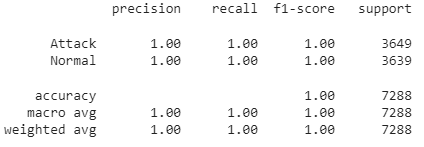
Figure 6 illustrates the Gradient boosting algorithm's classification accuracy. 

Figure 6: Classification accuracy of Gradient boosting classifier

The accuracy of the gradient boosting classifier was 99.72%. The attack and normal data prediction have an F1 score of 100% respectively. Both have perfect precision. The gradient boosting algorithm's confusion matrix is shown in Figure 7.

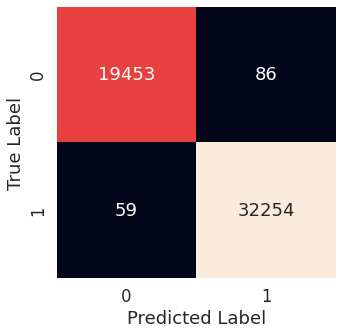
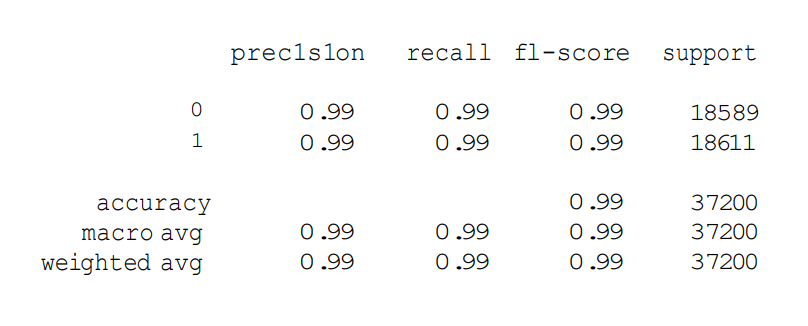


Figure 7: Confusion matrix of gradient boosting classifier

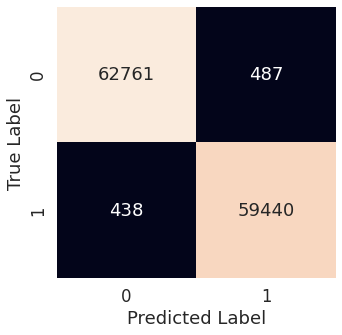
**Decision Tree**

Figure 8 depicts the classification accuracy of the decision tree method.



**Figure 8**: Classification accuracy of a decision tree classifier

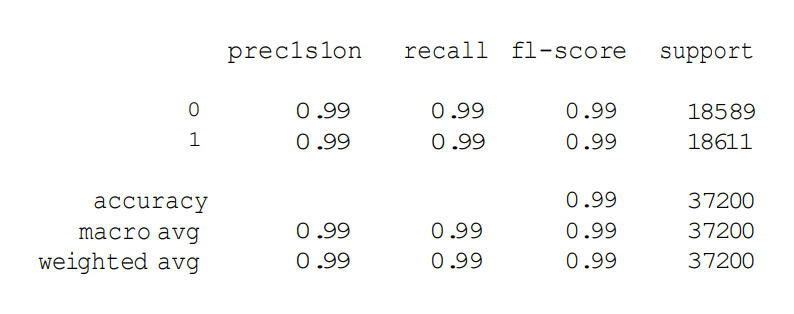
The accuracy of the decision tree algorithm was 99.24 percent. The attack and normal data prediction has an F1 score of 99% respectively. The decision tree algorithm's confusion matrix is shown in Figure 9.



**Figure 9**: Confusion matrix of a decision tree classifier

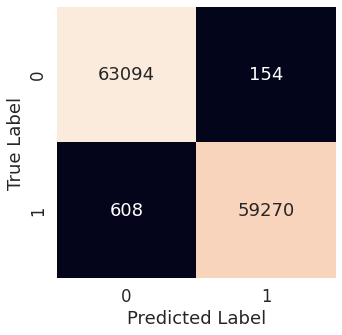
**Random Forest**

Figure 10 illustrates the random forest algorithm's classification accuracy.



**Figure 10:**  Classification accuracy of random forest classifier

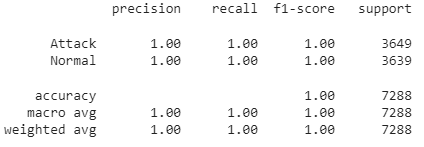
The accuracy of the random forest algorithm was 99 percent. The attack and normal data prediction has an F1 score of 99 %respectively. The random forest algorithm's confusion matrix is shown in Figure 11.



**Figure 11**: Confusion matrix of random forest classifier

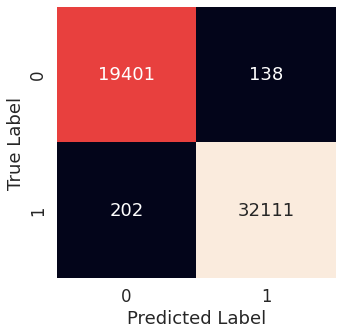
**Ada Boosting**

The classification accuracy of the ada boosting algorithm has been shown in figure 12.



**Figure 12**: Classification accuracy of ada classifier

The accuracy of the ada algorithm was 99.28 percent. For normal and attack, data the f1-score was 99%, respectively. The ada boosting algorithm's confusion matrix is shown in Figure 13.

****

**Figure 13**: Confusion matrix of ada classifier

The models are compared to those previously researched, as shown in Table 1. The Random Forest classifier 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 | 99.24 | Ref [21] BMA | 83.6 |
| Random Forest | 99.38 | Ref [11] Random Forest | 85 |
| Gradient Boosting | 99.72 | Ref [37] XGBoost | 91.23 |
| Ada Boosting | 99.34 | Ref [12] QSAR | 84 |

The accuracy rate of employed by the authors of [11] was only 85% accurate, the random forest classifier used in this investigation was 99% accurate. In this investigation, the DT classifier had the greatest attainable accuracy of 99%, whereas the BMA technique had an accuracy of 91.23 percent. [21] The accuracy of the authors' XG Boost was 94 percent. The Gradient Boosting approach was used in this work to achieve 99% accuracy, as opposed to the authors of [37], who used GB to achieve 94% accuracy. The Ada Boosting approach was used in this work to achieve 99% accuracy, as opposed to the authors of [40], who used Ada to achieve 81.28% accuracy

**CHAPTER 7**

**SKILLS**

**7.1 Introduction**

In this chapter we discuss the skills that we have obtained in order to develop this massive sophisticated system.

**7.2 Skills obtained**

Through this project the following skills have been developed:

* **Skill in Programming & Tools**
* **Anaconda Navigator**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

* **Python 3**

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.[30]Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.[31][32] Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0.[33] Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle-detecting garbage collection system (in addition to reference counting). Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020.[34]

**7.3 Summary**

In this chapter we discussed the list of skills that have been obtained throughout the process of developing and materializing this system.

**CHAPTER 8**

**FUTURE WORK**

**8.1 Introduction**

This chapter discusses the future scope or the implementation of this system. As our system is a web-based, various forms of new features can be incorporated to this system as per the requirements.

**8.2 Future Scope of Work**

Four machine learning techniques were provided in this work, and their comparative assessment is described. The goal of the article was to determine which machine learning classifier would be the most effective in predicting intrusion detection based on the dataset utilized. Fhour classifiers were built and their results were compared. Some of the comparison approaches used include the confusion matrix, accuracy, specificity, and sensitivity. . This kind of diagnosis will become more common in the future as machine learning algorithms improve as a result of continuous research. If additional patient information is utilized, the model may be refined and adjusted. A bigger dataset ensures more precise and accurate findings. This is critical since data lose is a very delicate problem that requires high degrees of accuracy and precision. In future we will implement different model with large dataset.

**8.3 Summary**

This chapter has described the possible future applications of the design. But there are a lot of possibilities with the designed system. The system may need some research for different applications, though the principle of the designed system will remain as it is.

**CHAPTER 9**

**DESIGN IMPACT**

**9.1 Introduction**

In this chapter, we discuss about the various impacts that our system has been able to generate.

**9.2 Environmental Impact**

By introducing this system in a center huge amounts of paper can be saved because all the tasks will be computerized therefore it would the number of paper wasted per day in official works can be hugely eradicated.

**9.3 Economic Impact**

The economic impact that this system entails is that by introducing this system in an organization, employees’ salary can be reduced significantly as there would now require less man power to complete a certain task because the system is now automated and requires less human effort.

**9.4 Social Impact**

This system will be socially acceptable as this kind of system is the need of the hour. In this era of ours, everything has been automated to provide comfort for the users. Therefore our system is no exception.

**9.5 Sustainability**

Our system has been able to deal with huge number of patients’ information at a time. When the numbers of medical tests are conducted simultaneously our system remains stable. Therefore based upon these facts and continuous testing, our system is sustainable.

**9.6 Summary**

This chapter has covered the different types of impacts that our system offers and those has been described and discussed. From the above given impacts we can conclude that our designed system is good enough to use under any circumstance.

**CHAPTER 10**

**COMPLIANCE WITH IEEE STANDARDS**

**10.1 Introduction**

In this section we discuss about the consistence of our task with diverse standards. There are a few distinct standards, amongst which the IEEE standards, US standards and European standards are talked about in this part.

**10.2 Compliance with IEEE standard**

There are a few distinct guidelines put forward by IEEE Standards affiliation. The majority of them however are not material for our framework. We have included idea of operation as for the IEEE standard.

**10.3. Compliance with US standard**

ANSI recommends that copyrighted software should only be included for informational purposes, or in forms which do not mandate particular implementations of the standard. Object code should never be included in a standard as a normative requirement. While ANSI opposes use of software standards to mandate particular implementations and believes that use of software in standards should be avoided to the extent possible, ANSI recognizes that there may be circumstances in which inclusion of some software, provided it is accompanied by adequate legal permissions, may facilitate development of multiple, competing and interoperable implementations of the standard. Examples of such software could include: ·

* Pseudo Code (code that is human readable and similar to programming languages but cannot be directly processed or compiled directly to be processed by hardware that manipulates data according to instructions);
* Schema examples;
* Data structure definitions;
* ASN.1 structure definitions;
* ABNF grammar specifications;
* Example programming instructions that are sufficiently limited in scope that they do not, either singularly or in the aggregate, perform a complete or a substantial part of a function and are illustrative, at most, of limited sections of an independent fully described specification; or
* Sample programming instructions provided solely for conformance testing purposes.

Our project has been established based on the above ANSI principles and it completely relies upon it.

**10.4 Summary**

In this section we have examined the different compliant standards and made sure that we are in accordance with. These standards have been put without hesitation so as to control things, guarantee well-being and ensure there are no well-being dangers to the use of distinctive segments. It is imperatively essential to maintain these measures and we have done as such over the span of our task work.

**CHAPTER 11**

**CONCLUSION**

Communication networks have always faced a significant challenge in terms of security. Software Defined Networks (SDN), a recent development in networking, has significant potential for providing highly secure communication networks maintains a centralized and global view of the entire network while decoupling the data and control planes. Networks that are feasible, robust, and secure because of this are now in place. Particularly recently, significant research efforts have focused on integrating SDN capabilities with intelligent traffic analysis by means of Machine Learning or Deep Learning. However, most efforts have consisted merely of transferring existing solutions into the SDN environment. At the time, security is a big issue for all enterprises and institutions' networks. Despite the development of many measures to assure the penetration of incursions into the network architecture, anomalies are trying to acquire successful access to these firms' data networks and Web services. Anomaly detection systems are designed to make it easier for computer systems to cope with threats. The creation of a machine learning model might aid in the early discovery of abnormalities and, as a result, reduce their severe repercussions. This research focuses on the effectiveness of numerous machine learning systems that incorrectly predict anomalies based on a set of numerical factors. A bigger dataset and machine learning models like Extra Trees Classifier and Voting Classifier might be used to improve the framework models in the future. The framework's consistency as well as its presentation will be improved as a result of this. Identifying and using these strategies for identifying new and developing threats will be the future of intrusion detection systems. The machine learning architecture may aid the general public in assessing the likelihood of a network intrusion happening in exchange for just providing some basic information. In a perfect world, it would help businesses and governments keep their most important information and money safe from hackers.

**BIBLIOGRAPHY**

[1] Z. Wang, Deep learning-based intrusion detection with adversaries. IEEE Access. 2018;6:38367–384

[2] S.M. Kasongo, Y. Sun. A deep gated recurrent unit based model for wireless intrusion detection system. Cakovec: ICT Express; 2020.

[3] J. Ribeiro, F.B. Saghezchi, G. Mantas, J. Rodriguez, R.A. Abd-Alhameed. Hidroid: prototyping a behavioral host-based intrusion detection and prevention system for android. IEEE Access. 2020;8:23154–168.

[4] N.T.T. Van, T.N. Thinh. Accelerating anomaly-based IDS using neural network on GPU. In: 2015 international conference on advanced computing and applications (ACOMP). IEEE; 2015. pp. 67–74.

[5] J. Jabez, B. Muthukumar. Intrusion detection system (IDS): anomaly detection using outlier detection approach. Procedia Comput Sci. 2015;48:338–46.

[6] S. Neelakantan, S. Rao. A threat-aware anomaly-based intrusion-detection approach for obtaining network-specifc useful alarms. In: International conference on distributed computing and networking. Springer. 2009; pp. 175–180

[7] S.M. Kasongo, Y. Sun. A deep learning method with flter based feature engineering for wireless intrusion detection system. IEEE Access. 2019; 7:38597–607.

[8] I. El Naqa, M.J. Murphy. What is machine learning? In: Machine learning in radiation oncology. Berlin: Springer; 2015. p. 3–11.

[9] S. Khatri, A. Arora, A.P. Agrawal. Supervised machine learning algorithms for credit card fraud detection: a comparison. In: 2020 10th international conference on cloud computing, data science & engineering (confuence), IEEE; 2020. pp. 680–83.

[10] P. Singh. Supervised machine learning. In: Learn PySpark. Springer; 2019. pp. 117–59.

[11] P. Harrington. Machine learning in action. New York: Manning Publications Co.; 2012.

[12] G. Dong, H. Liu. Feature engineering for machine learning and data analytics. Boca Raton: CRC Press; 2018

[13] N. Moustafa, B. Turnbull, K.K.R. Choo. An ensemble intrusion detection technique based on proposed statistical few features for protecting network traffic of internet of things. IEEE Internet Things J. 2018;6(3):4815–830.

[14] N. Moustafa, J. Slay. The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the KDD99 data set. Inf Secur J A Glob Perspect. 2016;25(1–3):18–31.

[15] C. Khammassi, S.A. Krichen GA-LR wrapper approach for feature selection in network intrusion detection. Compute Secure 2017;70:255–77.

[16] O. Osanaiye, H. Cai, K.K.R. Choo, A. Dehghantanha, Z. Xu, M. Dlodlo. Ensemble-based multi-filter feature selection method for DDOS detection in cloud computing. EURASIP J Wirel Commun Netw. 2016;2016(1):130.

[17] M.A. Ambusaidi, X. He, P. Nanda, Z. Tan. Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Trans Comput. 2016; 65(10):2986–98.

[18] B. Ingre, A. Yadav. Performance analysis of NSL-KDD dataset using ANN. In: 2015 international conference on signal processing and communication engineering systems, IEEE; 2015. pp. 92–6

[19] H. Alazzam, A. Sharieh, K.E. Sabri. A feature selection algorithm for intrusion detection system based on the pigeon-inspired optimizer. Expert Syst Appl. 2020;148:113249.

[20] Y. Deng, H. Duan. Control parameter design for automatic carrier landing system via pigeon-inspired optimization. Nonlinear Dyn. 2016; 85(1):97–106.

[21] T. Janarthanan, S. Zargari. Feature selection in UNSW-NB15 and KDDCUP’99 datasets. In: 2017 IEEE 26th international symposium on industrial electronics (ISIE). IEEE; 2017. pp. 1881–1886.

[22] V. Kumar, D. Sinha, A.K.Das, S.C. Pandey, R.T.Goswami. An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. Cluster Comput. 2020; 23(2):1397–1418.

[23] O. Almomani. A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms. Symmetry. 2020;12(6):1046

[24] https://www.kaggle.com/datasets/subhajournal/sdn-intrusion-detection

[25] G. Hossein and H. Hamid, "A new feature selection IDS based on genetic algorithm and SVM," 20 03 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7881798. [Accessed 15 11 2021].

[26] 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].

[27] 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].

[28] 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

[29] 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].

[30] "3.2.4.3.5. sklearn.ensemble.GradientBoostingClassifier — scikit-learn 0.20.3 documentation," Scikit-learn.org, 2009. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html. [Accessed 6 1 2022].

[31] V. Aliyev, "Gradient Boosting Classification explained through Python," Medium, 7 10 2020. [Online]. Available: https://towardsdatascience.com/gradient-boosting-classification-explained-through-python-60cc980eeb3d. [Accessed 6 1 2022].

[32] "sklearn.ensemble.AdaBoostClassifier — scikit-learn 0.22.1 documentation," scikit-learn.org, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html. [Accessed 6 1 2022].

[33] Great Learning Team, "AdaBoost Algorithm: Boosting Algorithm in Machine Learning," GreatLearning Blog: Free Resources what Matters to shape your Career!, 28 5 2020. [Online]. Available: https://www.mygreatlearning.com/blog/adaboost-algorithm/. [Accessed 6 1 2022].

[34] 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].

[35] 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.