Intrusion Detection System in Networks Employing a Double-Layer Architecture Using Machine Learning Algorithms

Amrit Mohapatra-222IT004
Information Technology
National Institute of Technology, Karnataka,Surathkal, India 575025
amrtimohapatra.222it004@nitk.edu.in

Natasha Jain-222IT023
Information Technology
National Institute of Technolgy, Karnataka, Surathkal, India 575025
natashajain.222it023@nitk.edu.in

Dr. Bhawana Rudra
Information Technology
National Institute of Technology, Karnataka, Surathkal, India 575025
bhawanarudra@nitk.edu.in

***Abstract*—Monitoring both the activities of the system itself and the traffic on the network is the job of an intrusion detection system, which is more commonly referred to by its acronym, IDS. The IDS works by analyzing the network traffic or system logs, looking for patterns and signatures of known threats, or deviations from normal behavior that may indicate an attack. This may be done in response to the activity or traffic being deemed suspicious or destructive. IDS can take many different shapes, but regardless of its appearance, their primary purpose is to identify potentially harmful traffic in a number of different ways. There are primarily two types of intrusion detection systems: those that monitor networks and those that monitor individual computers. The anti - virus program and the firewall are two examples of defensive mechanisms used by the intrusion detection system (IDS). IDS plays an important role in protecting computer networks and systems from various types of attacks, such as malware infections, hacking attempts, and unauthorized access. It alerts security administrators to potential threats, provide detailed information about the attack, and help to prevent or mitigate the damage caused by the attack.**

Keywords — network traffic, potential threats, malware infections, unauthorized access, firewall, intrusion detection system

I. INTRODUCTION

Protecting computer networks is becoming an increasingly crucial aspect of the data transfer process in today's linked society, where individuals from all over the world are connected to each other through the internet and numerous intranets. The network's security is a complicated issue that calls for a wide range of policies to be implemented in order to prevent and monitor illegal access, misuse, and alteration of resources that are available over the network. The procedure for protecting data can be greatly strengthened with the application of soft computing approaches. A device or software program is referred to as an intrusion detection system (IDS) when it is used to monitor the activities that are taking place on a network or within a system for any potentially harmful behaviors or breaches of policy. After that, the outcomes are manufactured, and they are transmitted to the management station. The IDS can detect potentially dangerous software in a number of different ways, the most important of which is by employing a number of different methodologies. In an effort to identify unauthorized access to a network, one sort of intrusion detection system known as a network intrusion detection system (NIDS) does an analysis of the data flowing over the network in search of indications that malicious software is being utilised.

One further way to characterize an intrusion is as the difficulty in identifying people who are abusing their legal access to a system by illegally utilising their computing capable device in a way that violates the terms of their authorization to use the system in the first place. To put it another way, an incursion is an issue caused by the illegal use of a computing capable device. The intrusion detection approach can be broken down into two primary categories: anomaly detection and abuse detection. In practise, the two categories are frequently combined. The capacity to discern patterns of inappropriate behavior is an essential component of not only spotting instances of misuse but also of spotting anomalies. In this paper, we propose double layer machine learning architecture model that can cover wide range of anomalies.

We have considered the following four machine techniques:

- Linear regression and Decision tree
- Naive bayes and Decision tree
- Linear regression and Xgboost
- Decision tree and Xgboost

These models seek to discover and classify odd actions and traffic in a network that may be harmful, with the end goal of increasing the network's level of security.

II. RELATED WORK

In the paper MLH-IDS [1] the authors offer a multi-level hybrid intrusion detection system that combines rule-based and machine learning techniques for efficient intrusion detection. They call their system a hybrid since it detects intrusions on many levels.The efficiency of the system was validated by the fact that it was able to attain high detection rates while maintaining low false positive rates. The difficulty with the article was that the rule-based system relies on established criteria and may not identify innovative assaults that do not fall into the specified categories. In the paper C.Lui el al. [2] developed a system of institution detection using scalable random forest K-means+ and deep learning approaches, in the proposed system. The suggested system is meant to overcome the constraints of existing systems, such as the inability to manage large-scale data and the lack of precise anomaly detection. These limitations were taken into account when designing the proposed system. The quality of the training data that was utilized for the deep learning model might potentially have an effect on the performance of the system.In another work The authors in the paper[3] present a unique hybrid intrusion detection system (IDS) that utilizes approaches from both random forest and modified NSGAII-ANN. The suggested system intends to increase the accuracy and efficiency of intrusion detection by simplifying the system's computational processes, with the goal of improving the process of feature selection and decreasing the complexity of the system.

By applying a dynamic threshold and integrating a number of different classifiers as described in TSE-IDS [4], the authors seeks to increase both the accuracy and the efficiency

of intrusion detection. They also present a dynamic threshold method with the purpose of lowering the false positive rate. This is accomplished by modifying the threshold in accordance with the degree of the anomalies that are found.

H. Yao et al [5] presented a framework that increased the accuracy and efficiency of intrusion detection by integrating numerous data mining techniques at different levels of abstraction. This was accomplished via the use of multiple data mining techniques. Network traffic, session, and application are the three levels of analysis that make up the framework. Different data mining approaches were utilized at each level in order for the authors to successfully extract characteristics and identify abnormalities. Unsupervised learning methods, such as K-Means and One-Class Support Vector Machines (SVMs), were utilized inside the framework that was developed for the purpose of feature extraction and anomaly detection.

T. Wisanwanichthan et al [6] suggested that a layered method be taken, which consists of two levels of classifiers: the first layer utilizing Naive Bayes to filter out regular data, while the second layer employs SVM to identify anomalous activity. The authors also presented a technique for selecting characteristics for intrusion detection that is based on Information Gain (IG) and Genetic Algorithm (GA). The goal of this method was to pick the features that were most important for detecting intrusions. A model for the detection of anomaly-based intrusions in Internet of Things backbone networks that use a two-layer dimension reduction and a two-tier categorization system. The first layer applied Principal Component Analysis (PCA), which minimized the dimensionality of the data. The second layer applies a two-tier classification model, which consists of a binary classifier and a multi-class classifier, to differentiate between normal and abnormal traffic.[7]Random Forest (RF), Gradient Boosting Decision Tree (GBDT), and K-Nearest Neighbors (KNN) are some of the machine learning techniques that are included in the novel hybrid approach to intrusion detection. Following the completion of feature selection using GBDT, the proposed method then moves on to classification with GBDT, KNN and RF, all working in conjunction with one another. A number of different machine learning classifiers such as K-Nearest Neighbors (KNN), Decision Tree (DT), and Support Vector Machine (SVM), were suggested by C. Shan [9] and colleagues. The model that is being presented makes a judgment by combining the predictions of many classifiers and then dynamically selecting the one that has the greatest performance for each new data point as it arrives.

The researchers T. Wisanwanichthan [10] and colleagues offer a double-layered hybrid technique for the detection of network intrusions. The first layer analyzes the incoming network traffic, and the second layer employs support vector machines to categorize the remaining data as either an attack or non-attack.The authors conduct an analysis of the suggested method by making use of the NSL-KDD dataset and contrast it with other methods that was considered to be state-of-the-art. Z. Yang et al. [11] proposed model extracts features from raw network traffic data using a stacked autoencoder (SAE), and then utilizes a fuzzy logic classifier to categorize those characteristics as either normal or abnormal. The model was assessed by utilizing the NSL-KDD dataset, and the findings demonstrate that the proposed model achieves high accuracy and beats many different state-of-the-art intrusion detection approaches. Due to the use of fuzzy logic, the benefit of the model that has been suggested is that it is able to successfully manage complicated and high-dimensional data while still maintaining its interpretability. Due to the fact that the model uses deep learning strategies, it necessitates a significant quantity of both training data and computational resources. This is one of the model's drawbacks.

In their study, Alzahrani et. al [12] present a hybrid strategy for detecting network intrusions that blends machine learning with rule-based approaches. This approach is known as the hybrid approach. The data on the network traffic are preprocessed by the proposed system using several strategies for feature selection. Following that, the Random Forest (RF) algorithm were used to categorize the traffic as either benign or malicious. In the event that the categorization is determined to be malicious, the system employs a rule-based mechanism to categorize the particular form of attack. The performance evaluation of the suggested system reveals that it has an accuracy of 99.23% and a false positive rate of 0.08%, which indicates that it has a high level of accuracy and a low level of false positives. The suggested method has a number of benefits, the most important of which are its high level of accuracy and its capacity to identify previously undiscovered threats. However, in order for machine learning algorithms to function well, the system needs a substantial quantity of labeled training data, and the rule-based approach might not be efficient enough to defend against sophisticated assaults.

The study by M. Ahmed [13] gives, a hybrid deep learning strategy for network intrusion detection systems (NIDS) is proposed. This approach combines long short-term memory (LSTM) and convolutional neural networks (CNN) to accurately categorize network traffic data as either normal or malicious. The suggested model was trained and tested using a dataset that was made accessible to the public, and it attained an accuracy rate of 99.79% as a result of these processes. The capacity of the suggested method to learn complicated and non-linear correlations in the data, as well as its high accuracy in identifying unknown threats, are two of the advantages of using this method. The most significant drawback, however, is the high level of computing expense required for both the training and deployment of deep learning models.

A hybrid machine learning strategy is presented in the paper written by P.-J. Chuang and S.-H. Li [14] for the purpose of network intrusion detection. In order to improve the effectiveness of network intrusion detection, the suggested method combines two existing algorithms: the k-nearest neighbor (k-NN) algorithm and the decision tree (DT) algorithm. In order to accomplish feature selection, the k-NN algorithm is utilized, while the DT algorithm is utilized in order to categorize network traffic as either normal or abnormal. The suggested method was tested on the NSL-KDD dataset, and the results show that it achieved a high detection rate while simultaneously maintaining a low false alarm rate. The technique is not suitable for use in scenarios with a large-scale network and more processing resources are needed. In a separate piece of research, the authors presented a hybrid machine learning strategy for the detection of network intrusions by combining genetic algorithm and decision tree techniques. The suggested method began by utilizing the genetic algorithm in order to improve the accuracy of the decision tree classifier's hyperparameter settings. After that, the improved decision tree is utilized to categorize the traffic on the network as either typical traffic or attack traffic.[15] On the NSL-KDD dataset, the method was examined, and the results showed that it obtained a detection rate of 99.63%

while having a false rate of 0.8%. The authors concluded that the technique that was suggested is capable of efficiently detecting network assaults while also decreasing the number of false positives.

The above mentioned papers have limited learning ability to detect new and unknown attacks. We are utilizing two layers so that even if the first layer misses an attack, the second layer may pick it up, ensuring that we detect high level of anomalies.

III. METHODOLOGY

A. *Proposed Methodology*

In this section we have proposed the working of algorithm. The algorithm is made of two layers, Layer 1 and Layer 2, which is depicted in Fig 1. In layer 1 we are using one machine learning algorithm that predicts whether it is as attack or not. In layer 2 the predictions which are misclassified by layer 1 are fed as input and again a different machine learning algorithm tries to find if it was attack or not. After that, we see how the anomaly-based IDS built on the Double Layered Approach can detect suspicious connections in real time. Layer 1 of our two-layer detection system is in charge of finding malicious activity. Finally, at the second layer, we have a specialized classifier designed to identify the threats missed by the initial model. Layer 1 and Layer 2 machine learning algorithms are chosen as per the analysis of results of each algorithm. The algorithms which give the best results are chosen.

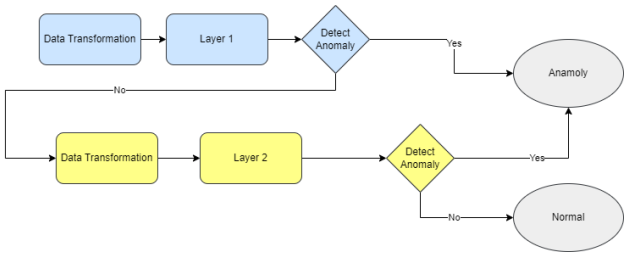


Fig.1: Flow Chart of Methodology

The below given algorithm1 represents an intrusion detection system that predicts a network connection is an attack or not. It uses double layer hierarchical architecture (DLHA).

Input: $X = \{f_1, f_2, \ldots, f_{40}\}$ **// 40 attributes captured**
Output: $y \in \{0, 1\}$
  while DLHA IDS is running **do**
    // for every network connection
    after performing data transform 1
    represent X_i as X_{I_1}
    if *Layer* 1 predicts X_{I_1} as 1 **then**
      $y \leftarrow 1$
      return y
    else
      Layer 2 is activated
      after performing data transform 2
      represent X_i as X_{I_2}
      if *Layer* 2 predicts X_{I_2} as 1 **then**
        $y \leftarrow 1$
        return y
      else
        $y \leftarrow 0$
        return y
      end if
    end if
  end while

Algorithm 1. Using Combined Model

B. *Dataset and Preprocessing*

The dataset consisting of numerous simulated intrusions in a military network environment has been provided. It created 0an environment to acquire raw TCP/IP packet data for a network by replicating a typical LAN used by the United States Air Force. Multiple assaults were directed at the LAN as if it were a real-world environment. Each connection is either labeled as normal or as an attack with a specific attack category. Each connection record is approximately 100 bytes long. From normal and attack data, for each TCP/IP connection, 41 quantitative and qualitative characteristics are extracted (3 qualitative and 38 quantitative features). The class variable contains two classes:

- Normal
- Anomalous

The dataset used for training consists 25192 rows and 42 columns and dataset used for testing has 22444 rows and 41 columns.

C. *Scaling Numerical Attributes*

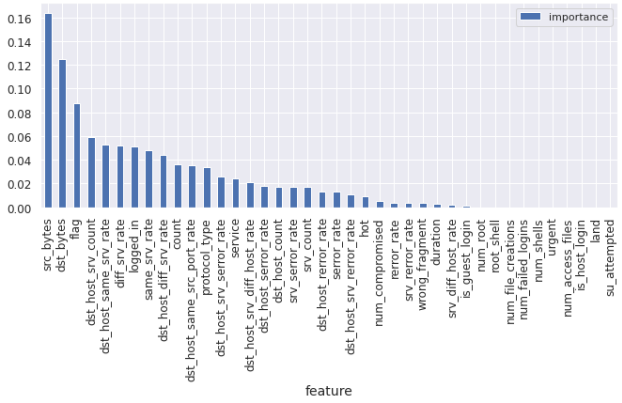
The process of extracting and normalizing the dataset's numerical columns happens during this stage. Rescaling the data to zero mean and a one standard deviation is the process of standardization. The data are then distributed normally with a standard deviation of one as a result. This is done to make sure that all of the numerical columns' values fall within the same range and that no one column is given preferential treatment. This is done in particular to guarantee that the values in each of the number columns fall within the same range. This is a crucial consideration because the performance of the machine learning algorithms depends on the standardization of the data.

D. *Encoding Categorical Attributes*

In order to get the data ready for machine learning models, we preprocess the data in a few different ways. It deals with two different kinds of data: numerical data, which comprises of numbers, and categorical data, which consists of categories (that consists of text or categories). The numerical data are the primary emphasis of the first section. It first pulls out all of the columns in the training and testing datasets that include numerical values, then adjusts those columns so that they have a zero mean and a one variance . This is done to ensure that all of the features have the same scale, which makes it simpler for machine learning models to learn from the data. Having all of the features have the same scale. Last but not least, the scaled data are saved in their own independent dataframes. The categorical data are the primary emphasis of the second section of this. It takes the datasets used for training and testing and extracts all of the columns that contain categorical values. After that, it converts these categorical categories into numerical values using an encoding algorithm. This is done because the models that are used for machine learning can only function with numerical values. At this point, the encoded data has been saved in its own independent dataframes, and the target column has been split out from the encoded data. In the process of getting the data ready for machine learning models, this preprocessing stage is quite important. It helps to increase the accuracy of the models by ensuring that the data is in a format that can be used by the models and ensuring that the data is in that format.

E. *Feature Selection*

The first step involves training the Random Forest Classifier for feature selection as shown in figure 2, using the data that has been provided in the training set. We are using RFC because the dataset is large, noisy, and has many features, it can handle missing values and outliers well. Upon the completion of the training phase, the 'importance's' data frame will have the significant features of the data set extracted and saved in it. When deciding how much weight to assign to each feature's contribution to the model's correctness, the model considers the accuracy of the model as a whole. After that, the significance of each feature is ranked in descending order, and then it is plotted as a bar graph so that a visual analysis can be performed to see which features have the most influence on the correctness of the model. This assists in identifying the aspects of the dataset that are most essential and can be used for feature selection in subsequent analyses. The following features have been chosen: dst bytes, src bytes, logged in, hot, count, same srv rate, dst host srv count, diff srv rate dst host same srv rate, dst host srv diff port rate, dst host diff srv rate.



a.

Fig2: Importance Graph of Features

IV. EVALUATING MODELS

A. *System Details*

The experiments were carried out in windows 10 64-bit Intel (R) Core (TM) i5-7200U CPU @ 2.70GHz (4 CPUs) processor with 8GB of RAM. Graphics card used was Intel (R) HD Graphics 620 memory of 4GB and NVIDIA GeForce 940MX of memory 2GB.

At first, we conducted experiments with conventional machine learning algorithms, and we selected those models that demonstrated a high level of accuracy in order to merge them and create a double layer architecture. Those algorithms include Naive bayes, K-Neighbors, XG Boost, Random Forest. For each model, we perform the following operations: To facilitate cross-validation, we divide the training data into 10 groups, or "folds." Adjusts the model so that it better matches the training data. We compute the cross-validation mean score, which is the model's performance measured over all folds on average. The accuracy of the model is determined which can be defined as the proportion of instances that were correctly categorized by the model when it was applied to the training data. This may be done by looking at the percentage of examples that were correctly classified. We compute the confusion matrix, which gives details of the number of cases that were correctly and mistakenly identified for each class. Then we generate a classification report that details the accuracy, recall, and F1-score of each class. Below are the results obtained from the traditional models. The experimental results are shown in fig3, fig4, fig5 and fig6. The performance evaluation criteria include:

- Cross Validation Mean Score: measure of the average performance of the model over all the cross-validation folds.
- Model Accuracy: measures the percentage of correctly classified instances in the dataset. Higher accuracy indicates better performance.
- Confusion matrix: a table that shows the number of true positives, false positives, true negatives, and false negatives.
- Classification report: provides summary of precision, recall, F1-score, and support for each class in the classification model

a) Support
b) Recall
c) F1- Score
d) Precision

```
Cross Validation Mean Score:
 0.9071666840303904

Model Accuracy:
 0.9071679709651809

Confusion matrix:
 [[7000 1245]
 [ 392 8997]]

Classification report:
              precision    recall  f1-score   support

     anomaly       0.95      0.85      0.90      8245
      normal       0.88      0.96      0.92      9389

    accuracy                           0.91     17634
   macro avg       0.91      0.90      0.91     17634
weighted avg       0.91      0.91      0.91     17634
```

Fig 3. Results of Naive bayes model

```
Cross Validation Mean Score:
 0.9914370153431007

Model Accuracy:
 0.9937620505840989

Confusion matrix:
 [[8168   77]
 [  33 9356]]

Classification report:
              precision    recall  f1-score   support

     anomaly       1.00      0.99      0.99      8245
      normal       0.99      1.00      0.99      9389

    accuracy                           0.99     17634
   macro avg       0.99      0.99      0.99     17634
weighted avg       0.99      0.99      0.99     17634
```

Fig 4. Results of K-Neighbors Classifier Model

```
Cross Validation Mean Score:
 0.9638761554915026

Model Accuracy:
 0.9652943177951684

Confusion matrix:
 [[8111  134]
 [ 478 8911]]

Classification report:
              precision    recall  f1-score   support

     anomaly       0.94      0.98      0.96      8245
      normal       0.99      0.95      0.97      9389

    accuracy                           0.97     17634
   macro avg       0.96      0.97      0.97     17634
weighted avg       0.97      0.97      0.97     17634
```

Fig 5. Results of XG Boost Classifier Model

```
Cross Validation Mean Score:
 0.9912666579204947

Model Accuracy:
 0.997448111602586

Confusion matrix:
 [[8224   21]
 [  24 9365]]

Classification report:
              precision    recall  f1-score   support

     anomaly       1.00      1.00      1.00      8245
      normal       1.00      1.00      1.00      9389

    accuracy                           1.00     17634
   macro avg       1.00      1.00      1.00     17634
weighted avg       1.00      1.00      1.00     17634
```

Fig 6. Results of Random Forest Classifier Model

V. VALIDATING THE PROPOSED COMBINED MODEL

In this study, we analyze the efficiency of our proposed machine learning algorithm that predicts anomalies in a dataset by combining the results of two different machine learning algorithms. The performance of various models is compared in terms of their ability to anticipate abnormalities in some data. We evaluate the effectiveness of two different machine learning models by contrasting how well they distinguish between "normal" and "anomalous" data. The first proposed double layered model is logistic regression model and decision tree model. The second proposed double layered model is Naive bayes and decision Tree. The third proposed double layered model is Linear Regression and XG Boost. The fourth proposed double layered model is Decision Tree and XG Boost. The performance of these models are shown in fig7, fig8, fig9, fig10, fig11, fig12, fig13, and fig14 respectively.

VI. EVALUATION AND RESULTS

A. *Evaluation Metrics*

1.Accuracy (all correct) =

$$\frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

2.Precision (predicted positives) =

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

3. Misclassification (all incorrect) =

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

4. Specificity (all actual negatives) =

$$\frac{TN}{TN + FP}$$

5. Sensitivity aka Recall (true positives) =

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

B. *Experimental Results*

1. Linear Regression and decision Tree

Table1. Experiment results of Linear Regression and Decision Tree double layered model.

Models / Evaluation Metrics	Traditional (Linear Regression)	Proposed method (Linear Regression and Decision Tree)
Accuracy	95.50	**99.76**
Misclassification	4.49	**0.23**
Precision	96.00	**99.74**
Sensitivity	94.16	**99.74**
Specificity	96.25	**99.77**

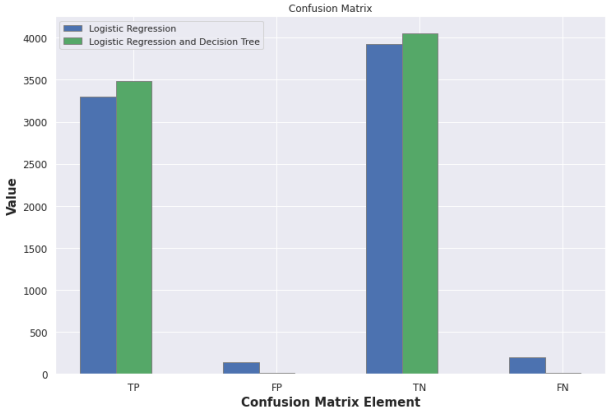


Fig7. Comparison of values

Fig8. Comparison of Performance

2. Naive bayes and decision Tree

Table2. Experiment results of Naïve bayes and decision Tree double layered model.

Models / Evaluation Metrics	Traditional (Naïve bayes)	Proposed method (Naïve bayes and decision Tree)
Accuracy	90.67	**99.70**
Misclassification	9.32	**0.29**
Precision	94.06	**99.62**
Sensitivity	85.22	**99.74**
Specificity	95.36	**99.67**

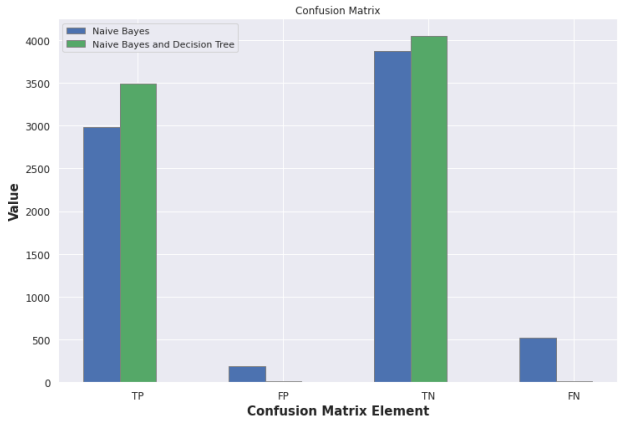


Fig9. Comparison of values

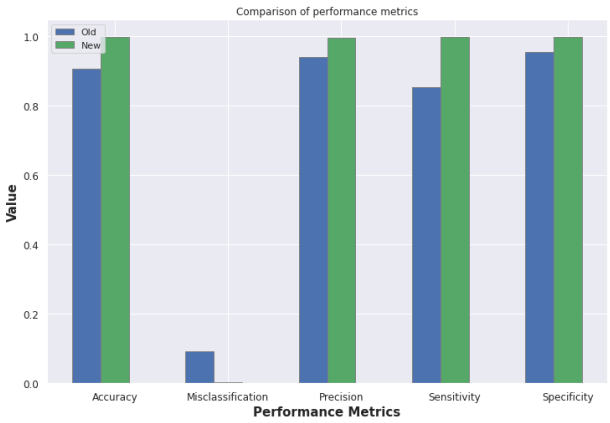


Fig10. Comparison of Performance

3. Linear Regression and XGBoost

Table3. Experiment results of Linear Regression and XGBoost double layered model.

Models / Evaluation Metrics	Traditional (Linear Regression)	Proposed method (Linear Regression and XGBoost)
Accuracy	95.50	**98.74**
Misclassification	4.49	**1.25**
Precision	96.00	**98.24**
Sensitivity	94.19	**99.05**
Specificity	96.62	**98.47**

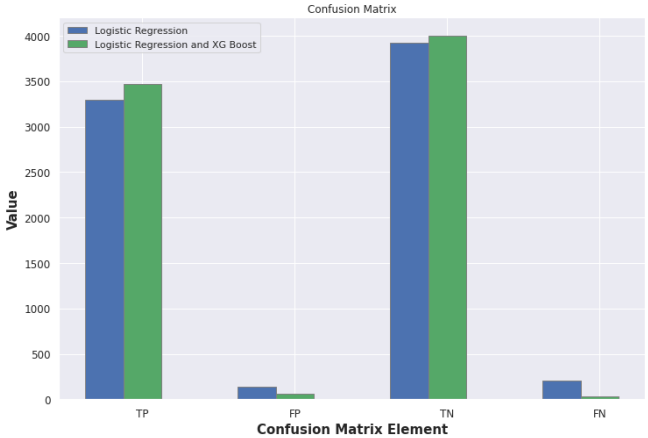


Fig11.Comparison of values

Fig.12 Comparison of Performance

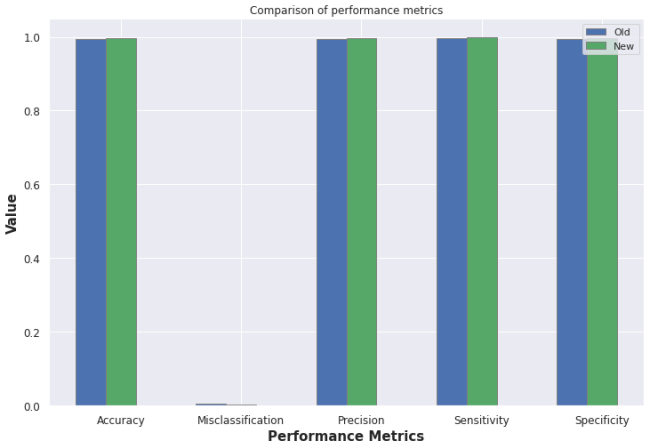


Fig.14 Comparison of Performance

4. Decision Tree and XGBoost

Table4. Experiment results of . Decision Tree and XGBoost double layered model.

Models / Evaluation Metrics	Traditional (Decision Tree)	Proposed method (Decision Tree and XGBoostt)
Accuracy	99.47	**99.72**
Misclassification	0.05	**0.02**
Precision	99.28	**99.54**
Sensitivity	99.57	**99.85**
Specificity	99.38	**99.60**

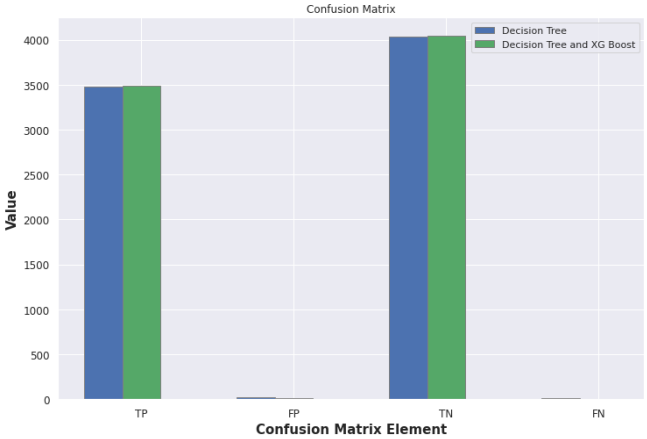


Fig13. Comparison of values

VII. Conclusion and Future Work

In the modern era of constantly growing connections worldwide, methods without layering are insufficient. Despite the fact that their performance is often quite promising, approaches to anomaly-based intrusion detection systems that use machine learning frequently suffer from bias towards regular attacks as well as an underestimation of unique hazards. Low detection rates, mainly for uncommon attacks, are caused by the inaccuracy in single machine learning models in classifying all types of assaults. This is especially true for attacks that are not very common. Thus, a layered approach is required to solve the IDS problem. A method known as a Double-Layered Approach (DLA) was proposed in this paper as a solution to the problem of inadequate performance on uncommon attacks. This method also resulted to an increase in the overall detection rate. Combining several different machine learning models is one way to cut down on irrelevant features and speed up the overall framework for real-time use. The portion that handles detection is split into two layers. The proposed double layered model that we developed was tested on the data set for the military network environment. It accomplished remarkable results, with an overall detection rate that was far higher than that of the conventional models. The execution time as well as the F1 score have demonstrated its increased productivity and capacity for a wider range of applications. The double layered IDS strategy has been shown to be successful and effective by the results of our experiments. As it makes use of two distinct classifiers. We arrived to the conclusion that our proposed double layered model gives a generalized model with a performance that leads the pack in recognizing assaults that are less common but more destructive after performing hyperparameter adjustment. This technique works well with a real-time intrusion detection system since it aims to secure important network settings. The application of this methodology to a network environment or data gathering that could classify assaults in a different way is one of the possible follow-up projects that could arise from this research.

Our future area of research will be to develop dynamic feature selection algorithms that adaptively select the most relevant features for intrusion detection in real-time, based on the current network traffic and attack patterns. Another challenge for intrusion detection systems is

scalability as attackers become more sophisticated. We will focus on developing algorithms that can handle large-scale networks and high-speed traffic without sacrificing accuracy. We will also investigate the use of adversarial attacks to evaluate the robustness of intrusion detection systems. Adversarial attacks could help identify weaknesses in the system and improve the system's ability to detect and respond to new and evolving threats.

VIII. REFERENCES

[1] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, ‘‘MLH-IDS: A multi-level hybrid intrusion detection method,’’ Comput. J., vol. 57, no. 4, pp. 602–623, Apr. 2014

[2] C. Liu, Z. Gu, and J. Wang, ‘‘A hybrid intrusion detection system based on scalable K-means+ random forest and deep learning,’’ IEEE Access, vol. 9, pp. 75729–75740, 2021.

[3] A. Golrang, A. M. Golrang, S. Yildirim Yayilgan, and O. Elezaj, ‘‘A novel hybrid IDS based on modified NSGAII-ANN and random forest,’’ Electronics, vol. 9, no. 4, p. 577, Mar. 2020

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

[5] H. Yao, Q. Wang, L. Wang, P. Zhang, M. Li, and Y. Liu, ‘‘An intrusion detection framework based on hybrid multi-level data mining,’’ Int. J.Parallel Program., vol. 47, no. 4, pp. 740–758, Aug. 2019.

[6] T. Wisanwanichthan and M. Thammawichai, "A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM," in IEEE Access, vol. 9, pp. 138432-138450, 2021, doi: 10.1109/ACCESS.2021.3118573.

[7] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K. R. Choo, ‘‘A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks,’’ IEEE Trans. Emerg. Topics Comput., vol. 7, no. 2, pp. 314–323, Apr./Jun. 2019

[8] Ü. Çavuşoğlu, ‘‘A new hybrid approach for intrusion detection using machine learning methods,’’ Appl. Intell., vol. 49, no. 7, pp. 2735–2761, 2019

[9] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, ‘‘An adaptive ensemble machine learning model for intrusion detection,’’ IEEE Access, vol. 7, pp. 82512–82521, 2019.

[10] T. Wisanwanichthan and M. Thammawichai, "A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM," in IEEE Access, vol. 9, pp. 138432-138450, 2021, doi: 10.1109/ACCESS.2021.3118573.

[11] T. Zhang, Z. Yang, Y. Guan, Y. Sun, and X. Liu, "A hybrid intrusion detection model based on deep learning and fuzzy logic," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 3, pp. 3181-3191, Mar. 2021.

[12] M. Alzahrani, S. Khan, and M. A. Alshehri, "A hybrid approach based on machine learning and rule-based methods for network intrusion detection," International Journal of Advanced Computer Science and Applications, vol. 11, no. 11, pp. 353-361, Nov. 2020.

[13] M. Ahmed, M. A. Ali, M. S. Hasan, and S. M. A. Motakabber, "A hybrid deep learning approach for network intrusion detection system," in Proceedings of the 2020 IEEE International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, Jan. 2020, pp. 1-6.

[14] P.-J. Chuang and S.-H. Li, "Network intrusion detection using hybrid machine learning," in Proceedings of the 2019 International Conference on Fuzzy Theory and Its Applications (iFUZZY), Hualien, Taiwan, Nov. 2019, pp. 1-5

[15] H. Gheibi, S. Shirazi, and S. Nezhadbandegani, "A hybrid machine learning approach for network intrusion detection using genetic algorithm and decision tree," International Journal of Network Security, vol. 20, no. 6, pp. 1187-1194, Nov. 2018