ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (11), November 2024, pp. 1-15

Enhancing Intrusion Detection with Cuckoo Search Optimized XGBoost Classifier for Network Traffic Analysis

K. Perachi

Assistant Professor, Department of Electronics and Communication Engineering, Sardar Raja College of Engineering, Tenkasi, Tamil Nadu, India. kperachi5@gmail.com

ABSTRACT

An Intrusion Detection System (IDS) is crucial for ensuring network security by identifying and mitigating malicious activities. With the rise in interconnected systems, vulnerabilities have become a significant concern, requiring advanced detection mechanisms. Machine learning techniques, with their ability to identify patterns and anomalies, offer a promising approach for anomaly-based IDS. This paper introduces a novel intrusion detection model combining the Cuckoo Search Algorithm (CSA) and the XGBoost classifier. This hybrid approach effectively analyzes network traffic and detects intrusions by leveraging the strengths of both methods. The model is trained and evaluated using the UNSW-NB15 dataset, achieving a commendable accuracy of 92.06%. Key contributions include handling missing values, outlier analysis, one-hot encoding, and feature scaling to enhance detection performance. The proposed model demonstrates superior results compared to existing techniques, offering an efficient solution for securing network infrastructures.

Keywords: Intrusion Detection Systems, Handling missing values and outliers, one hot encoding, Cuckoo Search Algorithm, XG Boosting classifier,

1 Introduction

Network Intrusion Detection (NID) systems, also known as Intrusion Detection Systems (IDSs), are a type of application that analyses network traffic or applications to find intrusive activities like unauthorized access, exploitation, and weaknesses generated by both external and internal attackers. Numerous technologies and techniques are created by an NID system, which keeps discrimination on network traffic and computer systems. [1] Specifically, Machine Learning (ML) technique detects patterns, abnormalities, or outliers related to a certain attack [2]. This will stop such attacks from occurring in the future process on cyber security. However, cyber security defences are less effective in detecting a zero-day attack [3]

A security tool called a Network Intrusion Detection System (NIDS) examines network data flows to identify dangers and stop malicious requests. [4]. This category has trouble spotting novel or zero-day attacks, but it successfully detects risks with low computation rates. [5]. Based on anomalies Conversely, NIDS has the benefit of detecting novel threats or odd behaviour, but it also has a high false positive rate. [6]. As a result, an NIDS that can accurately and frequently identify both new and known attacks must be designed. Using supervised, unsupervised, and reinforcement learning, Machine Learning has demonstrated promising outcomes when included into intrusion detection systems [7]. With a low rate of false positives

and a high degree of accuracy, supervised learning can detect more well-known attacks. Through methods like clustering or association, unsupervised learning has outperformed supervised learning in detecting zero-day attacks; nonetheless, the quantity of false positives has affectedly grown [8, 9].

Six well-known machine learning models such as random forest, decision trees, logistic regression, K-nearest neighbour, support vector machines, and Naive Bayes are exposed to be effective intrusion detection systems in industrial Internet of things surroundings. [10] Naive Bayes classification to classify the intrusion detection [11], ID intends a CNN-based method for anomaly-based intrusion detection systems in the Internet of Things, which successfully detects any possible intrusions and odd traffic patterns. [12] A two phase ID with Naive Bayes classifier is proposed to classify the data network traffic and an elliptic envelop method to detect anomaly, but the data network traffic is not identified because the envelop in the data network is not covered perfectly. [13] To improve the performance of IDS, Deep Learning (DL) techniques called CNN and RNN have been used widely however, they have yet to attain the degree of accuracy. [14, 15] When analysing network traffic, intrusion detection systems (IDS) able to identify the attack in the differentiate among malicious and legitimate traffic. The variety of network traffic characteristics and attack types in intrusion detection systems presents another challenge for DL. To overcome this problem Cuckoo Search Algorithm and XG boosting Classifier is used.

2 Related Work

Ramkumar *et al* (2022) [16] have proposed method uses a hybrid RSLO-tuned ensemble classifier in fog computing, with processes distributed across cloud, fog, and endpoint layers. It achieves high accuracy and improved precision through efficient feature selection and optimization techniques. However, the approach may face challenges with real-time processing due to computational complexity. Additionally, its effectiveness under diverse large-scale attack scenarios requires further evaluation.

Tait *et al* (2021) [17] have invented a Random Forest's binary classification and k-Nearest Neighbour for intrusion detection. These intrusion detection systems categorize binary and multiclass attacks based on whether the communication is deemed to be an attack or benign. The binary classification is more consistent than multiclass because it find the attacks constantly. The multiclass attacks yield the greatest accuracy when compared to the binary classification. Nevertheless, it is noted that the Random Forest accuracy and K-Nearest Neighbour accuracy is varying. The drawback for this technique is the accuracy value is not constant it varying in ranges.

Honghui Fan *et al* (2022) [18] have developed Synthetic Minority Oversampling Technique (SMOTE) Algorithm and Random Forest classifier to detect the network intrusion detection algorithm. To enhance the amount of minor samples and type of network attack a hybrid algorithm technique is used. This technique pooled the SMOTE sampling algorithm with K-

means clustering to evaluate the performance using the NSL-KDD dataset to get the classification accuracy. The disadvantage of this method is the type of the attack in not predicted correctly in the network.

Song *et al* (2023) [19] have proposed a Bidirectional Gated Recurrent Unit (BiGRU) and Temporal Convolutional Network (TCN) to classify the network traffic. The weights of features are not taken into account by TCN or BiGRU during feature extraction. Whereas BiGRU pulls overall temporal info from network traffic structures, TCN recovers local temporal information. Finally, is sent to the classifier, which uses it to categorize various types of network traffic. Its accuracy is predicted. The drawback is that in order to increase the model's capabilities, they reassign the weights of the temporal features.

Zhai *et al* (2023) [20] have developed a Convolutional Neural Networks–Gated Recurrent Units–Federated Learning (CNN–GRU–FL) to improve the model quality. A local training procedure based on CNN–GRU, an intrusion detection model improved the feature depiction capability by adding an attention mechanism. Furthermore, a trust-based node selection method is created to enhance Federated Learning's (FL) capacity for convergence. The effectiveness of this approach is founding global intrusion detection model across several independent organizations. To detect the CNN–GRU–FL training accuracy the convergence is captured. The drawback for this technique is low accuracy when compared to the proposed technique.

The contribution of this study is summarized as follows:

- The study evaluates the impact of dataset selection on the performance of intrusion detection systems, specifically using the UNSW-NB15 dataset.
- Introduces handling of missing values, outliers, and one-hot encoding to improve the reliability and quality of input data for network traffic analysis.
- Develops a Cuckoo Search Algorithm (CSA)-optimized XGBoost classifier to enhance intrusion detection accuracy and efficiency.
- Validates the proposed model using metrics like accuracy, precision, recall, F1-score, and AUC, demonstrating significant improvement over existing methods.

3 Proposed Method

This study's primary goal is to improve network intrusion detection models performance by implementing feature analysis and model optimization techniques. In order to increase the usefulness and robustness of the datasets, first data pre-processing is implemented. This includes one-hot encoding, and handing missing values and outliers to collect the missing data's. The pre-processed data is then subjected to a dual strategy of data analysis based on exploration and visualization.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (11), November 2024, pp. 1-15

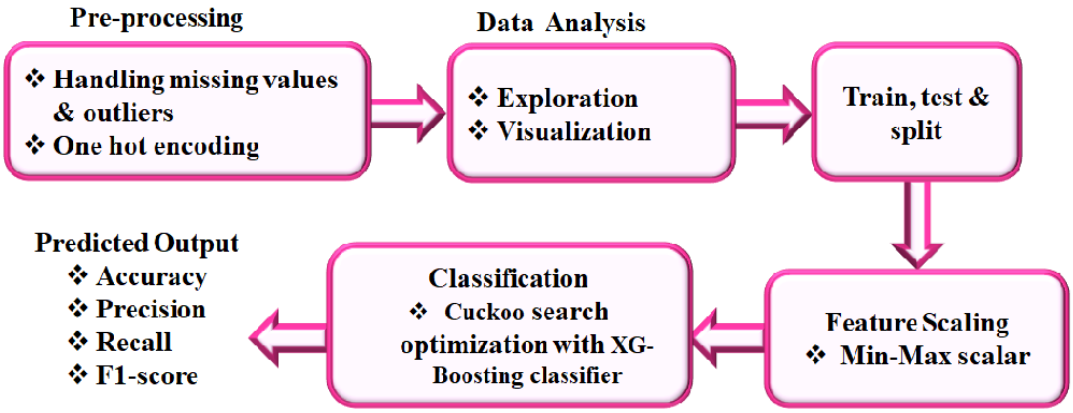

Figure 1: *Block diagram of proposed method*

Then exploration and visualization technique visualize a graph to detect the network traffic. After that the train and test is splited as 70% of train and 30% as test. Then feature selection based on mini-max scalar. Finally, test classifier verifying the performance using the technique as Cuckoo Search optimized XG boost classifier. The Cuckoo Search optimized XG boost classifier model is boosting the decision tree technique and then trained output is classified to get high accuracy and better network traffic in intrusion detection. The results are validated and evaluated using evaluation metrics like F1-score, accuracy, recall, and precision to get better accuracy compared to the existing systems.

3.1 Pre-processing

3.1.1 Handling missing values and outliers

Missing data and outliers are both important aspects of data in ML models. It is important to understand both of these models are accurate and reliable in ID. Missing data is a model to be less accurate because it takes fewer data to learn from input data. Outliers are a model to be less accurate because it skew the data and cause the model to learn from incorrect data. Removing or fixing inconsistent, unnecessary, or noisy data is known as data cleaning. This procedure involves addressing outliers, handling missing values, and resolving data inconsistencies. Cleaning the data helps to ensure the quality and reliability of the input data for intrusion detection.

3.1.2 One-hot encoding

One-hot encoding is used to convert categorical features that aren't quantitative into a numerical representation on computer-based algorithms. The statistical techniques and mathematical models that support machine learning algorithms usually work with numerical data. But, discrete symbols or labels that don't have any direct numerical value type of categorical features. To solve this problem one-hot encoding allows encoded data features by expanding the distinct feature values into a Euclidean space. Now, a different binary variable is assigned

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (11), November 2024, pp. 1-15

to each conceivable feature value. For every feature, a new binary variable is used, with the dualistic bit linked to the feature fixed to 1 and the other dualistic bits fixed to 0. This results in the conversion of category characteristics into numerical values. It is an essential supervised learning pre-processing step for the structured dataset.

3.2 Data analysis

3.2.1 Exploration and visualization

Exploration utilizes statistical methods and visualization in the first stage of data analysis to comprehend the nature of the data. Intrusion detection visualization aids a security administrator in intuitively identifying anomalous activity. Due to the intuitive nature of intrusion detection, visualization facilitates quick analysis and reaction. A network-based visualization technique for intrusion detection uses a visual graph to represent the source, destination, port, and other information of the network's packets. Therefore, this technique identifies an intrusion when an attack deviates from the graph's usual state and extract the attack's diagnostic properties to implement intrusion detection. Nevertheless, these techniques display audit data rather than notifications directly.

3.3 Train-Test-Split

It split the training dataset into training and testing sets with a 70-30 split ratio, the code uses the 'train-test-split' function. This division makes it easier to train the model on a portion of the data and then assess it on the data that isn't visible. Splitting the data makes it possible to evaluate the model's generalization performance, and standardization is essential for algorithms that are sensitive to the scale of input features. The train–test split method is used to divide the dataset into training and testing. The following is how the split dataset appears:

$$x\_train, \_x\_test, y\_test, y\_train \tag{1}$$

3.4 Feature Scaling

3.4.1 Min-max Scaler

The min-max scalar has two stages of normalization techniques; the preliminary step in the normalizing process is to minimize the excessive variation in the same feature's data values. In this stage, the logarithmic function is applied to the various feature values when there is a significant discrepancy between the feature's maximum and minimum values.

If $(X_{Max} - X_{Min} > threshold$

Then $$X_{normalize} = \log(x) \tag{2}$$

Where,

X represent the value of feature trying to normalize

The Min/Max Scalar approach, which involves determining the maximum and minimum of every data feature and then relating the subsequent function, is used in the second normalization step,

$$X_{normalize} = x - X_{Min}/X_{Max} - X_{Min} \quad (3)$$

Where x is the value of the feature we are attempting to normalize, X_{Min} is the smallest value the feature x observed, and $X_{Max}$is the highest value the feature observed. To get a better update of weights during the learning phase, the goal of this phase is to acquire all values between 0 and 1.

3.5 Classification by Cuckoo Search Optimized XG Boost

3.5.1 XG Boost

XGBoost is generated primarily to use Gradient Boosted decision trees for speediness and performance. Utilizing all available memory and hardware resources for tree boosting techniques is made easier with XGBoost. Another name for XGBoost is extreme Gradient Boosting. . Its benefits include enhancing algorithms, optimizing models, and being able to be integrated into computer systems. Stochastic Boosting, Gradient Boosting, and Regularized Boosting are the 3 main gradient boosting methods that XGBoost is accomplished of executing. The algorithm sorts the greatest use of memory assets and is very effective in reducing computation time. It takes the distinct ability to boost on extra data that has already been contributed to the trained model, allows parallel structure in tree construction, and capable of handling missing information.

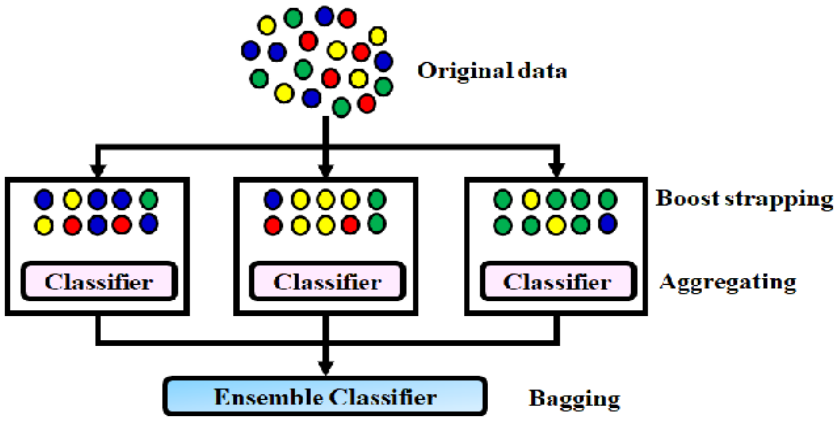

Figure 2: *Basic XG boosting classifier*

Because it addressed the mainstream of the issues that arose from the previous models, XGBoost is selected. XGBoost is well-suited to identify and handle missing values. At this time, resultant to the condition at the layer, the data splits up into aggregating. To refine the parameters of XG Boost Classifier the proposed work introduces CSA.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (11), November 2024, pp. 1-15

3.5.2 Cuckoo Search Algorithm (CSA)

In 2009, Xin-She Yang and Deb created the meta-heuristic technique known as CSA, It is employed to resolve difficult issues. It is triggered by cuckoos' natural behaviour, particularly the mandatory breeding parasitism of certain species, which involves depositing their eggs in other host birds' nests to increase their chances. Two sets of rules are used to define the CSA algorithm, and they are as follows:

1. Every cuckoo lay an egg at the same time step and transfers it into an individually selected nest.
2. The higher quality of eggs will be transferred to produce further generations,

The total number of host nest is stable, and the likelihood of an egg laid by a bird can be identified over crowd bird (0, 1). In this case, the host either takes those eggs out of the nest or starts a new hive from scratch. The best path from the font to the end server is found using the meta-heuristic technique known as CSA.

Algorithm: CSA

Input: Server attributes that are taken into account during the process

Output: Best servers

i. Create preliminary cuckoo population for n amount of servers

ii. Calculate average energy consumption $Avg\ EC$ of all servers

iii. For every Server s, f $EC_s \geq Avg\ EC, EC_s = Avg\ EC$

iv. While($t < ITR$)//t : amount of counts

v. Choose a cuckoo at random

vi. Calculate fitness of the particular cuckoo on the base of energy consumption

vii. To catch best solution $Bsol$ built on fitness task

viii. End (While)

ix. Return: $Bsol$

x. End (CS)

The population is generated in Step (i) based on the number of servers. In step (ii), out how much energy each server uses on average. In step (iii), the energy consumption of each server develops the average energy consumption if it is higher than or equal to the average energy consumption. To determine the optimal solution based on the fitness function, steps (iv-vii) are repeated. Using the algorithm the energy consumption is best in intrusion detection system.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (11), November 2024, pp. 1-15

Article Title: Enhancing Intrusion Detection with Cuckoo Search Optimized XGBoost Classifier for Network Traffic Analysis

3.5.3 Evaluation metrics

Using various performance metrics the proposed outline to training data is evaluated. The evaluation metrics are accuracy, precision, recall, F1 score and AUC. Using the following equation each metrics are defined as

i) Accuracy: From the predicted data accuracy measures the amount of suitably recognized samples. The formula for calculating accuracy is to divide the total quantity of samples (TS) by the sum of true positives (TP) and true negatives (TN).

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

ii) Precision: By dividing TP by the sum of TP and false positives (FP) computation is achieved. The ratio of accurately (TP) to the sum of projected positive samples (AD) is called precision.

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

iii) Recall: TP is divided by the sum of TP and FN to find the ratio of TP to the total number of ID in the dataset.

$$Rcall = \frac{(TP)}{(TP+FN)} \quad (6)$$

iv) F1 score: A combined measure of precision and recall is called as F1 score.

$$F1 - score = \frac{(2*\text{Precision}*\text{Recall})}{(Precision + Recall)} \quad (7)$$

At last, the simulation performance across different classifications is evaluated using the AUC, which describes the connection between the true positive and false positive rates.

4 Result and discussion:

4.1 Dataset

The Australian Centre for Cyber Security's (ACCS) Cyber Range Lab released the UNSW-NB15 dataset, which blends modern synthetic attack characteristics with real-world typical network activity. The dataset, which was recorded in a private setting, includes attack flows and benign flows. The number of records is 25192 in that the training set of 17634 samples and the testing set of 7558 samples to find as normal and anomaly class. The class label is one of 42 features that define each flow in the dataset. In both test and train sets each row in the 42nd column is labeled as binary either normal or anomaly. Using this featured targeted value and the classified value, the confusion matrix is plotted.

4.2 Protocol-type distribution

In protocol-type distributions 3 protocol type are used they are tcp, udp and icmp it have the count ranges from 0 to 20000. Where tcp stances for Transmission Control Protocol and its counts ranges from 0 to 20000 which means from sender to receiver, udp stances for User

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (11), November 2024, pp. 1-15

Article Title: Enhancing Intrusion Detection with Cuckoo Search Optimized XGBoost Classifier for Network Traffic Analysis

Datagram Protocol, its ranges from 0 to 2500 it is used for communication purpose and icmp stances for Internet Control Message protocol it ranges from 0 to 1500 its function is network management.

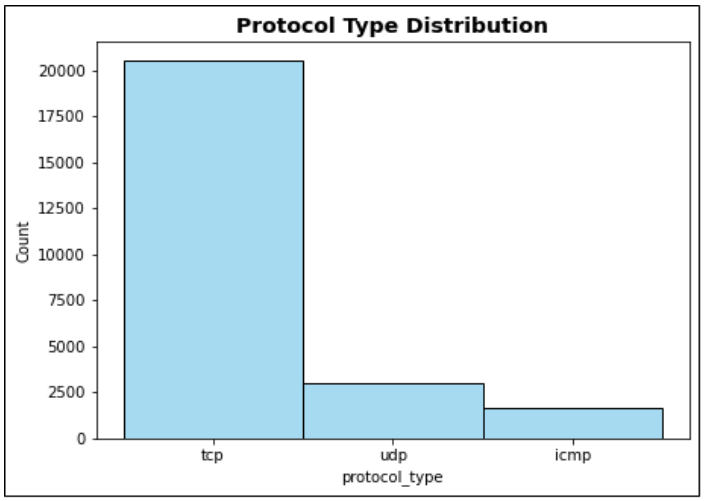

Figure 3: *Protocol Type Distribution*

Figure 4 shows the top 5 flags horizontal bar plots SF, SO, REJ, RSTR and RSTO which has the count ranges from0 to 14000. Flag count varies for each 5 flags as SF ranges from 0 to 14500, SO ranges from 0 to 7000, REJ ranges from 0 to 3000, RSTR ranges from 0 to 800 and RSTO ranges from 0 to 300 respectively.

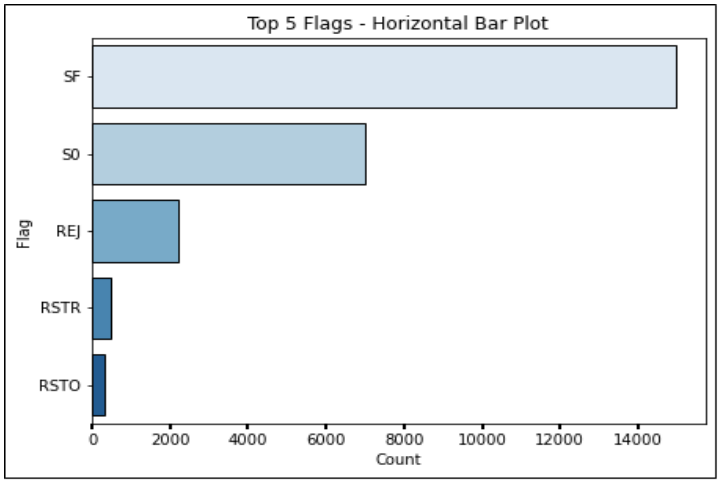

Figure 4: *Top 5 flags- horizontal bar chart*

The pie chart distribution differentiates the classes in to normal and anomaly where classes have the percentage of 60% anomaly and 40% normal. The anomaly means irregular or abnormal values in the network traffic, normal values means stable or regular values in the network traffic is displayed in figure 5.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (11), November 2024, pp. 1-15

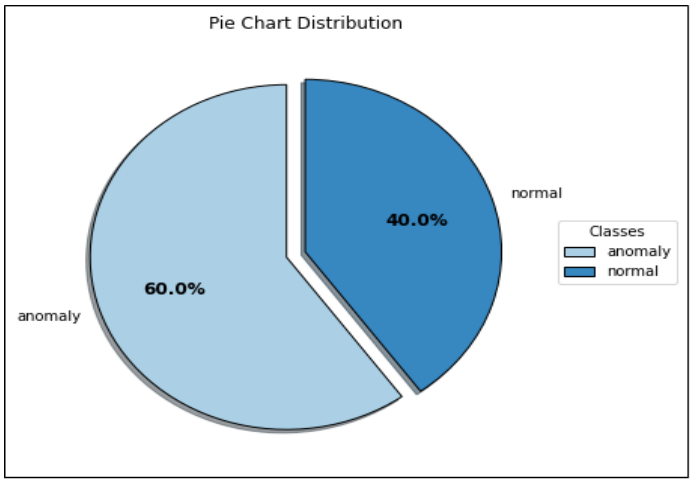

Figure 5: *Pie Chart distribution*

Whereas class distribution classes as normal and anomaly have the count of 0 to 14000. Anomaly ranges from 0 to 13000 the classes where not normal in this range and normal ranges from 0 to 11500 the classes where average in this stage.

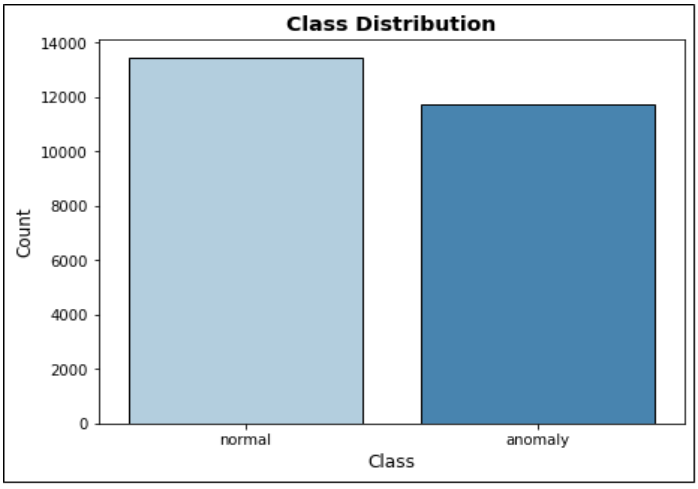

Figure 6: *Class Distribution*

DoS by class have both service and count is to feature the normal and anomaly using the tcp, udp and icmp protocol. The count ranges from 0 to 60000 where the tcp ranges from 0 to 10000 at that time it is normal and from 10000 to 10100 it is anomaly in the network traffic, udp is anomaly from 0 to 500 and the icmp is normal in the range of 0 to 100. DoS by class is shown in figure 7.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (11), November 2024, pp. 1-15

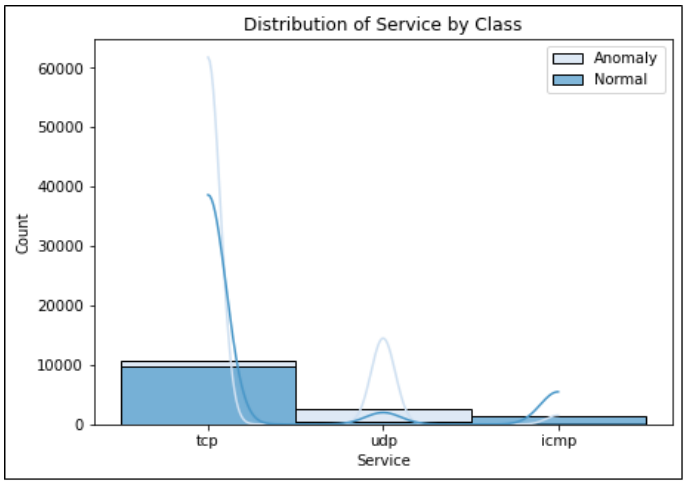

Figure 7: *Distribution of service by class*

4.3 Model loss and model accuracy

The measures were the main basis for evaluating the model loss and model accuracy using the UNSW-NB15 datasets.

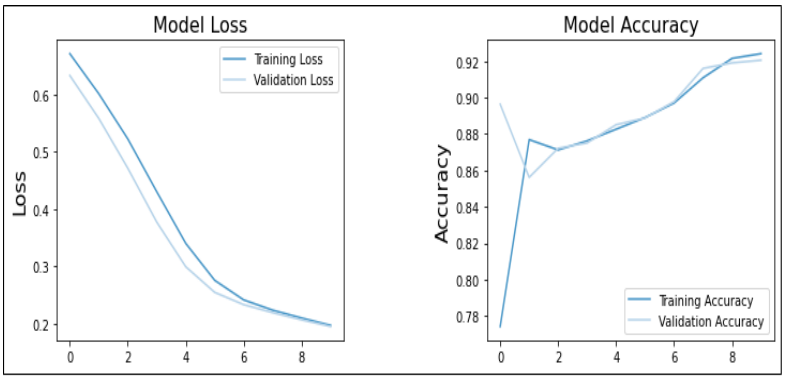

Figure 8: *Model loss and model accuracy*

The accuracy value is trained using the XG boosting with CSA to optimize the model to get the higher accuracy of 92.06%. When compared to the existing method.

4.4 Confusion Matrix

According to the prediction false positives and false negatives values are predicted in confusion matrix. This indicates that both anomaly and normal flow situations are detected by the model with a high and reliable detection rate. The confusion matrix shows the results of true label vs predicted label.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (11), November 2024, pp. 1-15

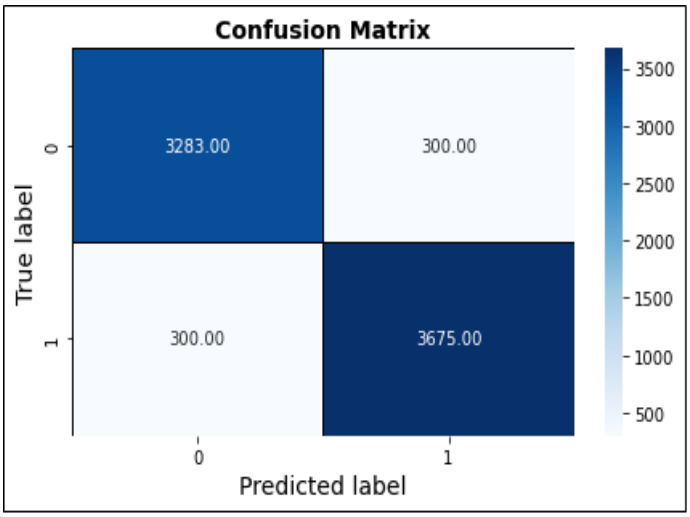

Figure 9: *Confusion Matrix*

4.5 ROC curve

The ROC curve is curved using true positive rate and false positive rate, then the detection rates (recall) for identifying anomaly and normal flows varied from 92.45%. ROC curves are used in the figure 10 to show in a way that accurately the model detects flows on datasets. The F1-score values of different flows and their allocation counts in the test dataset were used to calculate weighted F1 metrics.

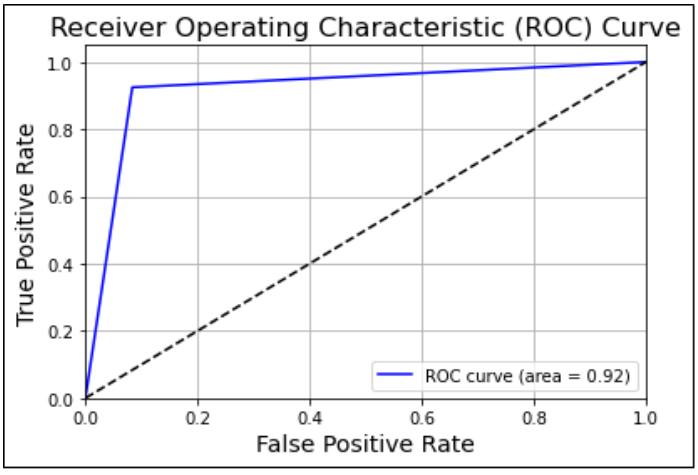

Figure 10: *ROC Curve*

4.6 Comparison

In contrast, several other studies [20, 21] ignored the time series component of the training data and only used baseline data. There was no training data for feature analysis in these studies and their results in table 1. Where the CNN-RGRU accuracy is 78.79%, multiclass classifier accuracy is 80.3% and the proposed method XG boosting accuracy is 92.06%.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
Vol. 03 (11), November 2024, pp. 1-15

Article Title: Enhancing Intrusion Detection with Cuckoo Search Optimized XGBoost Classifier for Network Traffic Analysis

Table 1: *Comparison for accuracy*

Methods	Datasets	Accuracy
CNN-GRU	NSL-KDD	78.79%
Multi class classifier (SHAP)	NSL-KDD	80.3%
XG Boosting classifier	UNSW-NB15	92.06%

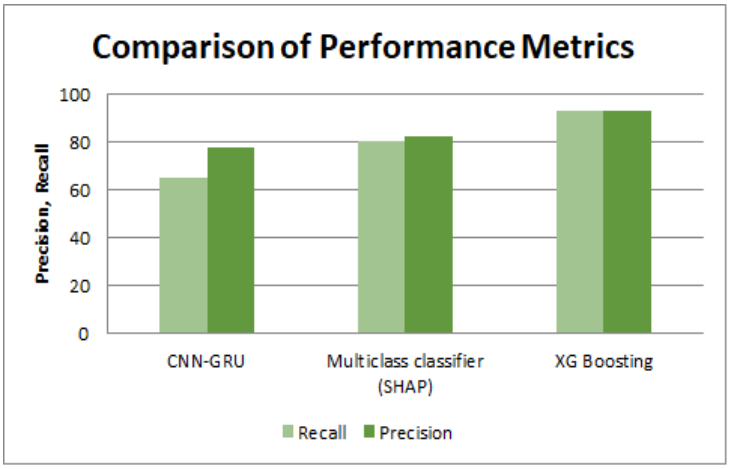

Figure 11: *Comparison of performance metrics*

5 Conclusion

This paper presents a novel approach to intrusion detection in network security using a CSA-optimized XGBoost classifier. The proposed method effectively handles pre-processing challenges, including missing values and outliers, while leveraging advanced feature scaling techniques to enhance model performance. The UNSW-NB15 dataset was utilized for evaluation, achieving a notable accuracy of 92.06% and demonstrating superior performance compared to existing models like CNN-GRU and multiclass classifiers. This work highlights the potential of combining optimization techniques and ensemble classifiers for robust and efficient anomaly detection in modern network infrastructures. Future work can explore further improvements to handle diverse attack scenarios and reduce computational complexity.

References

1. Ghanbarzadeh; Reza; Ali Hosseinalipour; Ali Ghaffari, Year: 2023, “A novel network intrusion detection method based on metaheuristic optimisation algorithms”, Journal of ambient intelligence and humanized computing, Vol: 14, no: 6, pp. 7575 – 7592.

2. Ridwan; Mohammad Azmi; Nurul Asyikin Mohamed Radzi; Fairuz Abdullah; Y. E. Jalil, Year: 2021, “Applications of machine learning in networking: a survey of current issues and future challenges”, IEEE access, Vol: 9, pp. 52523 – 52556.

3. Ahmed; Imran; Marco Anisetti; Awais Ahmad; Gwanggil Jeon, Year: 2022, "A multilayer deep learning approach for malware classification in 5G-enabled IIoT", IEEE Transactions on Industrial Informatics, Vol: 19, no: 2, pp. 1495 – 1503.

4. Yang Zhen; Xiaodong Liu; Tong Li; Di Wu; Jinjiang Wang; Yunwei Zhao; Han Han, Year: 2022, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection", Computers & Security, Vol: 116, pp. 102675.

5. Masdari; Mohammad; Hemn Khezri, Year: 2020, "A survey and taxonomy of the fuzzy signature-based intrusion detection systems", Applied Soft Computing, Vol: 92, pp. 106301.

6. Dwivedi; Shubhra; Manu Vardhan; Sarsij Tripathi; Alok Kumar Shukla, Year: 2020, "Implementation of adaptive scheme in evolutionary technique for anomaly-based intrusion detection", Evolutionary intelligence, Vol: 13, no: 1, pp. 103 – 117.

7. Nguyen; Thanh Thi; Vijay Janapa Reddi, Year: 2021, "Deep reinforcement learning for cyber security", IEEE Transactions on Neural Networks and Learning Systems, Vol: 34, no: 8, pp. 3779 – 3795.

8. Alimi; Oyeniyi Akeem; Khmaies Ouahada; Adnan M. Abu-Mahfouz; Suvendi Rimer; Kuburat Oyeranti Adefemi Alimi, Year: 2021, "A review of research works on supervised learning algorithms for SCADA intrusion detection and classification", Sustainability, Vol: 13, no: 17, pp. 9597.

9. Almalawi; Abdulmohsen; Adil Fahad; Zahir Tari; Asif Irshad Khan; Nouf Alzahrani; Sheikh Tahir Bakhsh; Madini O. Alassafi; Abdulrahman Alshdadi; Sana Qaiyum, Year: 2020, "Add-on anomaly threshold technique for improving unsupervised intrusion detection on SCADA data", Electronics, Vol: 9, no: 6, pp. 1017.

10. Eid; Abdulrahman Mahmoud; Bassel Soudan; Ali Bou Nassif; MohammadNoor Injadat, Year: 2024, "Comparative study of ML models for IIoT intrusion detection: impact of data preprocessing and balancing", Neural Computing and Applications, Vol: 36, no: 13, pp. 6955 – 6972.

11. Li Yanan; Tao Qin, Yongzhong Huang; Jinghong Lan; ZanHao Liang; Tongtong Geng, Year: 2022, "HDFEF: A hierarchical and dynamic feature extraction framework for intrusion detection systems", Computers & Security, Vol: 121, pp. 102842.

12. Zhao; Ruizhe; Yingxue Mu; Long Zou; Xiumei Wen, Year: 2022, "A hybrid intrusion detection system based on feature selection and weighted stacking classifier", IEEE Access, Vol: 10, pp. 71414 – 71426.

13. Vishwakarma; Monika; Nishtha Kesswani, Year: 2023, "A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection", Decision Analytics Journal, Vol: 7, pp. 100233.

14. Saba; Tanzila; Amjad Rehman; Tariq Sadad; Hoshang Kolivand; Saeed Ali Bahaj, Year: 2022, "Anomaly-based intrusion detection system for IoT networks through deep learning model", Computers and Electrical Engineering, Vol: 99, pp. 107810.

15. Silivery; Arun Kumar; Ram Mohan Rao Kovvur; Ramana Solleti; LK Suresh Kumar; Bhukya Madhu, Year: 2023, "A model for multi-attack classification to improve intrusion detection performance using deep learning approaches", Measurement: Sensors, Vol: 30, pp. 100924.

16. Ramkumar; M. P., T. Daniya; P. Mano Paul; S. Rajakumar, Year: 2022, "Intrusion detection using optimized ensemble classification in fog computing paradigm", Knowledge-Based Systems, Vol: 252, pp. 109364.

17. Tait; Kathryn-Ann; Jan Sher Khan; Fehaid Alqahtani; Awais Aziz Shah; Fadia Ali Khan; Mujeeb Ur Rehman; Wadii Boulila; Jawad Ahmad, Year: 2021, "Intrusion detection using machine learning techniques: an experimental comparison", In 2021 International Congress of Advanced Technology and Engineering (ICOTEN), IEEE, pp. 1 – 10.

18. Wu Tao; Honghui Fan; Hongjin Zhu; Congzhe You; Hongyan Zhou; Xianzhen Huang, Year: 2022, "Intrusion detection system combined enhanced random forest with SMOTE algorithm", EURASIP Journal on Advances in Signal Processing 2022, no: 1, pp. 39.

19. Song Yangyang; Nurbol Luktarhan; Zhaolei Shi; Haojie Wu, Year: 2023, "TGA: a novel network intrusion detection method based on TCN, BiGRU and attention mechanism", Electronics, Vol: 12, no: 13, pp. 2849.

20. Zhai Feng; Ting Yang; Hao Chen; Baoling He; Shuangquan Li, Year: 2023, "Intrusion detection method based on CNN–GRU–FL in a smart grid environment", Electronics, Vol: 12, no: 5, pp. 1164.

21. Wang Maonan; Kangfeng Zheng; Yanqing Yang; Xiujuan Wang, Year: 2020, "An explainable machine learning framework for intrusion detection systems", IEEE Access, Vol: 8, pp. 73127 – 73141.