Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 17 No. 15 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijim.v17i15.37969

PAPER

A Network Intrusion Detection Approach Using Extreme Gradient Boosting with Max-Depth Optimization and Feature Selection

Ghassan Muslim Hassan[1], Abdu Gumaei[2](✉), Abed Alanazi[2], Samah M. Alzanin[2]

[1]Computer Science Department, College of Science, Mustansiriyah University, Baghdad, Iraq

[2]Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia

a.gumaei@psau.edu.sa

ABSTRACT

Network intrusion detection system (NIDS) has become a vital tool to protect information and detect attacks in computer networks. The performance of NIDSs can be evaluated by the number of detected attacks and false alarm rates. Machine learning (ML) methods are commonly used for developing intrusion detection systems and combating the rapid evolution in the pattern of attacks. Although there are several methods proposed in the state-of-the-art, the development of the most effective method is still of research interest and needs to be developed. In this paper, we develop an optimized approach using an extreme gradient boosting (XGB) classifier with correlation-based feature selection for accurate intrusion detection systems. We adopt the XGB classifier in the proposed approach because it can bring down both variance and bias and has several advantages such as parallelization, regularization, sparsity awareness hardware optimization, and tree pruning. The XGB uses the max-depth parameter as a specified criterion to prune the trees and improve the performance significantly. The proposed approach selects the best value of the max-depth parameter through an exhaustive search optimization algorithm. We evaluate the approach on the UNSW-NB15 dataset that imitates the modern-day attacks of network traffic. The experimental results show the ability of the proposed approach to classifying the type of attacks and normal traffic with high accuracy results compared with the current state-of-the-art work on the same dataset with the same partitioning ratio of the test set.

KEYWORDS

classification, extreme gradient boosting, feature selection, machine learning, network intrusion detection, optimization

1 INTRODUCTION

Since cyberattacks are growing at an alarming rate [1–3], network intrusion detection is a significant research area. In order to suggest important strategies for thwarting

Hassan, G.M., Gumaei, A., Alanazi, A., Alzanin, S.M. (2023). A Network Intrusion Detection Approach Using Extreme Gradient Boosting with Max-Depth Optimization and Feature Selection. *International Journal of Interactive Mobile Technologies (iJIM)*, 17(15), pp. 120–134. https://doi.org/10.3991/ijim.v17i15.37969

Article submitted 2023-01-09. Resubmitted 2023-05-21. Final acceptance 2023-06-18. Final version published as submitted by the authors.

malicious damage-causing cyber activity, several studies and approaches have been placed forward. However, when cyber-attacks escalate in complexity, the current solutions fall short of solving the issue. Due to the sophistication of cyber-attack vectors, traditional and simple defensive strategies such as firewalls, authentication, and antivirus appear to be ineffective [4]. The application of classification algorithms can be used to solve the significant decision-making problem of network intrusion detection [5, 6].

In the field of network intrusion detection, several machine learning algorithms have been used, including fuzzy logic, neural networks, support vector machines, Naive Bayes, K-nearest neighbors, and decision trees [7]. The performance of different procedures can be improved whenever an ensemble approach or a combination of several classifiers is used. A well-known machine learning strategy is the ensemble paradigm, which uses various algorithms to enhance predictions. The use of the ensemble paradigm has also been shown in some studies to be useful in increasing detection accuracy and speed [8–10].

In the same context, the suggested method emphasizes the use of a supervised machine learning concept to introduce a classification approach based on a notion known as stacking or stacked generalization. Stacked generalization is beneficial because it relies on merging predictions from various single classifiers, which can also significantly enhance the generalization, as shown in [11–13]. The benefit of stacking was discussed in the context of protein classification, and the desired accuracy was attained [14]. Classifier ensembles or committees provide better results than individual classifiers by controlling the bias and variance more skillfully [15–17].

The performance of SVM was compared to that of classification methods such as J48, AdaBoost, random forest, logistic regression, and BayesNet. It was obvious that every combination method that used SVM gave results that were superior to those of using SVM alone [18]. Using the MAWILab dataset, the super learner ensemble learning algorithm was implemented, producing better predictions [19]. Stacking is an example of an ensemble learning paradigm that takes into account multiple machine-learning methods, uses a meta-model to merge prediction results from various algorithms, and subsequently increases performance. The effectiveness of the detection process can be improved by combining the benefits of various algorithms [20]. The stacking technique has been used to identify malware on mobile devices, and the accuracy and F1-measure have been improved [21].

2 RELATED WORK

In order to develop applicable network intrusion detection using effective machine learning algorithms, several approaches have been proposed by researchers. An overview of these strategies that aim to boost performance overall is provided in this section. The use of neutrosophic-based logic classifier, which is an extension of fuzzy logic, has been proposed as a new approach for network intrusion detection, containing an ensemble design. To create the rules, a genetic algorithm was employed. In comparison to other methods, the aforementioned design might reduce the false-alarm rate to 3.19% [22].

A well-known classifier called the support vector machine (SVM) can classify from a small number of samples while still making the best predictions [23]. Using an intrusion detection dataset collected by a basic security module (BSM) for auditing data of the Defense Advanced Research Projects Agency (DARPA), Chen et al. [24] showed that SVM outperformed the artificial neural network (ANN)-based model in terms of detection rate. This is because SVM can achieve better through

comparatively fewer data and can perform faster than ANN, which needs a lot of training data. SVM is known to be particularly good at a binary classification task, but when used in conjunction with the other classifiers, it can also produce better classification results for a multiclass classification task.

Combining two separate models can result in better performance, as was shown by an ensemble strategy using multilayer perceptron with radial basis functions. The hybrid model developed by Govindarajan and Chandrasekaran [25] appeared more accurate when compared to individual models. The University of New Mexico created a dataset for this study that included both typical and unusual mail application trace data.

A system for intrusion detection was created by combining SVM and K-nearest neighbors (KNN). An ensemble design that achieved an enhancement of 0.756% in accuracy as compared with practical swarm optimization (PSO) was created using weights generated by PSO [26]. An adaptive intrusion detection strategy was created by Rangadurai Karthick et al. [27] by fusing hidden Markov and naive Bayesian methods.

According to experimental findings, the combinative analysis approach mentioned above produced successful outcomes and effectively learned the nature of network traffic. The hybrid model was implemented using traces from the DARPA and Center of Applied Internet Data Analysis (CAIDA) datasets. To lessen the bias typically found in class-wise predictions, a two-step hybrid approach for a binary classification task using KNN was proposed. In this method, KNN is utilized to categorize those instances whose classes are left undetermined in step one, while step one is used for binary classification and the additional module to identify abnormal cases [28].

Malik et al. [29] proposed a hybrid intrusion detection method that uses random forest (RF) and binary particle swarm optimization (BPSO) to categorize probe attack patterns. The effective search optimizer BPSO and the effective classifier RF both helped to improve performance. Eight other classifiers were compared to this method, and it was interesting to see that the BPSO-RF combination outperformed the individual classifiers. Zhou and Cheng [30] proposed an ensemble model using C4.5 and random forest with forest penalizing attributes (FPA).

The average of probability (AOP) algorithm was used in this study to combine the results of various classifiers using the cutting-edge intrusion detection dataset CICIDS2017. Results showed an extremely positive increase in accuracy of 96.76%. By combining a genetic algorithm with decision trees, Khammassi and Krichen [31] were able to conduct an insightful study in which the genetic algorithm served as a search strategy and the decision trees served as a classification tool. Using the UNSW NB-15 dataset, it was found that this method had an accuracy rate of 81.42% and a false alarm rate of 6.39%. Rajagopal et al. [32] used a two-class neural network to implement the binary classification task of network intrusion attacks. They demonstrated the importance of several combinations of UNSW NB-15 dataset features. Their results concerning to feature reduction achieved 97% of accuracy using 23 features. Gudla et al. [33] proposed a framework for attack detection using a long short-term memory deep leaning (LSTMDL) model. The authors found that the LSTMDL model achieved 94.11% of accuracy on UNSW-NB15 dataset divided into 80% for training and 20% for testing. Recently, Kumar and Sharma [34] used a multilayer perceptron (MLP) architecture with a grid search technique for intrusion classification. They achieved a competitive performance result up to 98.10% of accuracy; however, the MLP network architecture may be stuck in the local minimum, and it may not be able to boost its accuracy over a particular threshold. Moreover, MLP network architecture is sensitive to the randomization of the initial weight values.

3 PROPOSED APPROACH

The approach starts with loading the values of the network traffic extracted features. The methods of the research approach are given in Figure 1 and described in the following subsections. The output of the approach is the class label of the network traffic, which are Abnormal or Normal class label.

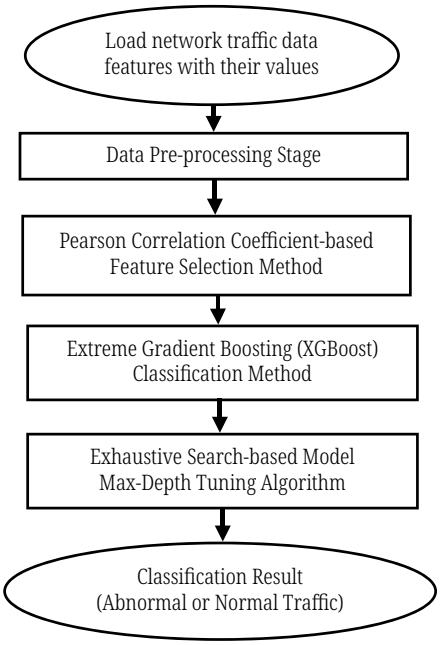

Fig. 1. The flowchart of the research approach methods

3.1 Data pre-processing method

The data pre-processing method of the proposed approach is required and it includes three main steps: data cleaning, data type conversion, and normalization. In data cleaning, the data with null values are dropped. Data type conversion maps the values of categorical features to numerical values using one-hot-encoded. Because the features have values with different ranges and may affect the accuracy of the model, the normalization step is necessary. The values of the feature are normalized using a min-max scaler to be in the range of [0, 1].

3.2 Pearson correlation coefficient-based feature selection method

The process of choosing which features to use in model construction is known as feature selection. These features must be consistent, non-redundant, and relevant. As the size of dataset features and their variety continue to increase, it is crucial to reduce the size systematically. Improving the performance of predictive models and lowering computational modeling costs are two main objectives of feature selection. Numerous techniques can be applied to feature selection, and they can be categorized into three groups: filtered techniques, wrapper techniques, and embedded techniques [35]. Independent of the selected predictor, the subsets of variables are chosen in the filtered methods as a pre-processing step. The embedded methods perform variable selection during the training process and are typically specific to given

learning machines, whereas the wrapper methods use the learning machine of interest as a black box to score subsets of variables according to their predictive power. The Pearson Correlation Coefficient (PCC)-based filtered method is used in the proposed approach because it is quicker than wrapper and embedded methods and robust against over-fitting, which could introduce bias [36]. Pearson's correlation coefficient is a statistical model, represented by r value. It returns a value that represents the degree of correlation for any two variables [37]. Taking the covariance between the features and the predictor and dividing it by the sum of their standard deviations yields the Pearson correlation coefficient. The change of scale in the features does not affect the coefficient. The output of the method contains the features and their scores, as ranked by predictive power. The scores are in the range between 0 and 1 and a threshold value between them can be used as a filter to select the best features.

3.3 Extreme gradient boosting (XGBoost) classification method

Extreme Gradient Boosting (XGB) is an ensemble learning algorithm proposed by Chen Tianqi [38]. It uses quadratic Taylor expansion in the loss function and adds a regularization term to make the model simpler and reduce overfitting. XGB can automatically use the multi-threading of the central processing unit (CPU) for parallelism, and at the same it can process high-dimensional sparse features in a distributed manner, making it more accurate and faster than similar algorithms for a wide range of applications [39]. The XGB algorithm is an improvement of the gradient-boosting decision tree (GBDT), which can be used for both classification and regression problems [40]. It is considered a scalable machine learning algorithm of the boosted trees, which is often used in some large-scale data features with remarkable effect and can perform large-scale parallel boosted tree algorithm operations efficiently, flexibly, and conveniently. The parameters of the XGB method are divided into three categories, namely general parameters, boost parameters, and learning parameters. For boosting the model, it is recommended to optimize the boost parameters of the algorithm. The implementation process of the XGB is written in Algorithm 1.

Algorithm 1: The Implementation Process of the XGB

Input: training samples $X = \{(x_1, y_1), (x_2, y_2),..., (x_m, y_m)\}$, a maximum number of iterations; regularization coefficients, max-depth, and other hyper-parameters.
Output: strong learner $f(x)$;
Begin
For t=1, 2, ..., T do the following steps
Step 1:
- Calculate the loss function L of the ith sample ($i = 1, 2,.., m$) in the current round based on the regularization coefficients, max-depth, and other hyper-parameters.
- Calculate the first and second derivatives of all samples.

Step 2:
- Try to split the decision tree based on the score of the current node, the default score value is 0. The score of the node that needs to be split currently is the sum of the first and second derivatives.

Step 3:
- Split the sub-number based on the division feature and eigenvalue corresponding to the maximum value.

Step 4:
- If the maximum value is 0, the current decision tree is established, and the weights of all leaf regions are calculated to obtain the weak learner.
- Update the strong learner and enter the next iteration of the weak learner.
- If the maximum value is not 0, continue from step 2 to try to split the decision tree.

End

In the proposed approach, the XGB method is used for classifying normal and abnormal networks as an effective model of the network anomaly detection tool.

3.4 Exhaustive search-based model max-depth tuning method

This subsection explains the method used to find the best value of XGB's max-depth hyper-parameter using an exhaustive search (ES) algorithm [41]. The ES algorithm is a problem-solving brute-force search technique that systematically travels all possible solutions in a specific search space instead of randomly guessing all of these possible solutions. It is often applied when the search space is discrete. In our case, the max-depth will have a value from a range from 1 to n. Algorithm 2 states the pseudo code to accomplish the steps of max-depth optimization.

Algorithm 2: Max Depth Optimization of XGB Using ES

Input: training samples $X = \{(x_1, y_1), (x_2, y_2),...,(x_m, y_m)\}$, test samples $Y = \{(x_1, y_1), (x_2, y_2),...,(x_m, y_m)\}$, the maximum number of iterations T, regularization coefficients, and other hyper-parameters, the maximum number of max-depth N.
Output: strong learner $f_j(x)$ with maximum Acc_j;
Begin
For j=1, 2, ..., N do the following steps
Step 1:
For t=1, 2, ..., T do the following steps
Step 1.1:
- Calculate the loss function L of the ith sample ($i = 1, 2,.. m$) in the current round based on the regularization coefficients, max depth j, and other hyper-parameters.
- Calculate the first and second derivatives of all samples.

Step 1.2:
- Try to split the decision tree based on the score of the current node, the default score value is 0. The score of the node that needs to be split currently is the sum of the first and second derivatives.

Step 1.3:
- Split the sub-number based on the division feature and eigenvalue corresponding to the maximum value.

Step 1.4:
- If the maximum value is 0, the current decision tree is established, and the weights of all leaf regions are calculated to obtain the weak learner.
- Update the strong learner and enter the next iteration of the weak learner.
- If the maximum value is not 0, continue from step 1.2 to try to split the decision tree.

Step 1.5:
Assign the strong learner $f(x)$ to a list of strong learners $f_j(x)$
Step 2:
$Acc_j \leftarrow$ Compute the accuracy of $f_j(x)$ on the test set.
End

4 EXPERIMENTS AND DISCUSSION

This section presents the experiments of the study to show the performance of the proposed approach to detect normal and abnormal network traffic from the selected features. The experimental results will be evaluated using a holdout technique in which the dataset is divided into two sets: one set for training and the other set for testing. Here, the ratio of 80:20 will be used in the experiments. All experiments are implemented and coded on a laptop core i7 with CPU 2.20GHz and using Python programming language. The following subsection describes the dataset, evaluation measures, and experimental results.

4.1 Dataset

The UNSW-NB-15 dataset is used to validate the proposed approach. It was created in 2015 by Moustafa and Slay [42] in the Cyber Range Lab of UNSW Canberra and they argued that it has a better representation of the modern nature of network attacks. The dataset contains a huge number of network raw packets produced using the IXIA PerfectStorm tool through generating a hybrid of synthetic modern attack behaviors and real normal activities [43]. Also, the tcpdump tool was applied to obtain 100 GB of raw traffic stored in PCAP files. The UNSW-NB-15 dataset has nine forms of attacks, namely, Analysis, Fuzzers, Backdoors, Exploits, DoS, Generic, Shellcode, Reconnaissance, and Worms. The Bro-IDS and Argus tools are utilized, and twelve algorithms are established to create a total of 49 features including two class labels with 175341 rows. The first-class label is for normal and abnormal activities and the second-class label is for the types of attacks. This study selects the first-class label to detect anomalies in the network traffic as a binary classification task. After dropping non-important features and rows with null values, the dataset contains 45 features and 81173 rows. Table 1 shows these features with their data types. Features are categorized into five groups: time features, content features, basic features, flow features, and additional generated features. The number of instances and the distribution of normal and abnormal classes in the dataset are shown in Table 2 and Figure 2, respectively.

Table 1. Dataset features with their data types

Data Type	Feature
Object	service, proto, attack_cat, state
Integer	id, dpkts, spkts, dbytes, sbytes, dttl, sttl, sload, dload, swin, dtcpb, stcpb, dwin, dmean, smean, response_body_len, trans_depth, ct_state_ttl, ct_srv_src, ct_src_dport_ltm, ct_ftp_cmd, ct_dst_ltm, ct_flw_http_mthd, ct_dst_sport_ltm, ct_src_ltm, ct_srv_dst, ct_dst_src_ltm
Float	sloss, rate, dur, dloss, dinpkt, sinpkt, tcprtt, sjit, djit, ackdat, synack
Binary	is_ftp_login, is_sm_ips_ports, label

Table 2. The number of instances in the dataset

Class Label	Number of Instances
Abnormal	61685
Normal	19488
Total	81173

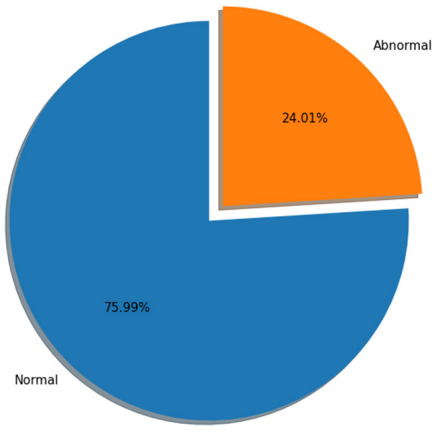

Fig. 2. The distribution of normal and abnormal classes in the dataset

4.2 Evaluation measures

A number of evaluation measures such as precision, recall, accuracy, and F1-measure are calculated from the classification results. These measures are obtained using the equations listed below:

$$Percision = \frac{TP}{(TP + FP)} \tag{1}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{2}$$

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{3}$$

$$F1 - measure = 2 * \left(\frac{Recall * Precision}{Recall + Precision} \right) \tag{4}$$

where *TP*, *FP*, *FN*, and *TN* are the true positive, false positive, false negative, and true negative cases, respectively.

The above measures are computed based on the confusion matrix. The confusion matrix of classification is employed to obtain the number of true positives and false negatives in the test set of network traffic instances. For precision, true and false positives are classified accurately by dividing the number of true positives by the number of true positives and adding the number of false positives. For recall, the completeness of true positives and false negatives is given by dividing the number of true positives by the number of true positives and adding the number of false negatives. The *F*1-measure merges the recall and precision of classifier's results into a single metric using Eq. (4). In addition, the accuracy is another metric gives the fraction of correct classification results for the model. Mathematically, it is computed by the number of instances, which are correctly classified divided by the total number of instances.

4.3 Evaluation results

After applying the proposed approach methods to the experimental dataset, the data pre-processing method ensures that the dataset is cleaned by checking the null values of the features. If some rows have null values, they will be removed. Also, the data pre-processing method converts the values of categorical features into numerical values using the one-hot-encoding technique and normalizes all feature values to be in the range of 0 and 1. Then, the Pearson correlation coefficient-based feature selection method is used in which the features with correlation coefficient scores more than or equal to 0.2 with the target class label are selected. Table 3 presents the selected features with their correlation coefficient scores.

Table 3. Selected features with their correlation coefficient scores

Feature	Correlation Coefficient Score
ackdat	0.203839
service_ftp-data	0.212381
service_ssh	0.224627
service_dns	0.225843
swin	0.227784
dwin	0.227784
proto_tcp	0.227784
proto_udp	0.227784
state_FIN	0.228613
dmean	0.240989
sload	0.334562
dload	0.343910
rate	0.344535
ct_src_ltm	0.368486
ct_dst_ltm	0.387358
ct_src_dport_ltm	0.444874
ct_srv_dst	0.459984
ct_srv_src	0.463153
ct_dst_src_ltm	0.463735
ct_dst_sport_ltm	0.497234
state_INT	0.546631
state_CON	0.552505
sttl	0.707337
ct_state_ttl	0.801403

To train and test the classification and optimization methods of the approach, the dataset is divided randomly into a train set (80%) and a test set (20%) for training and testing tasks. Table 4 shows the number of instances in the train and test sets.

Table 4. A number of instances in the train and test sets

Class Label	Number of Instances		Total
	Train Set	Test Set	
Abnormal	49406	12279	61685
Normal	15532	3956	19488
Total	64938	16235	81173

During the experiments, we trained the XGB classifier with the default values of its hyper-parameters on the test set. Figure 3 presents the confusion matrix containing the number of examples that are correctly classified.

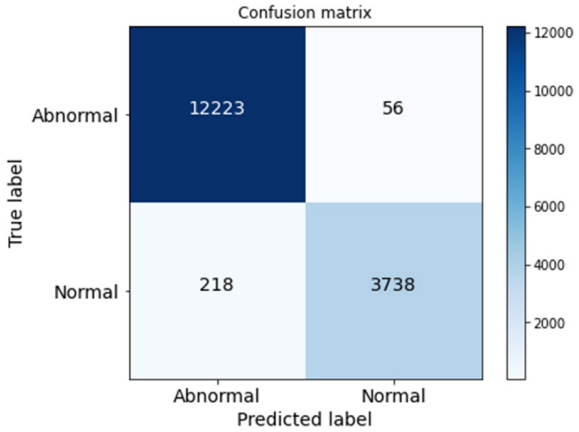

Fig. 3. Confusion matrix of XGB classifier on the test set using the default values of hyper-parameters

From the results of the confusion matrix in Figure 3, Table 5 displays the results of the other evaluation measures. It shows that the XGB achieves 98.31% of accuracy, 0.9832 of weighted average precision, 0.9831 of weighted average recall, and 0.9830 of weighted average F1-measure. Also, we can see that the model attains 0.9839 of macro average precision, 0.9702 of macro average recall, and 0.9768 of macro average F1-measure.

Table 5. The experimental results of the proposed approach using the default values of XGB hyper-parameters

Class Label	Precision	Recall	F1-Measure
Abnormal	0.9825	0.9954	0.9889
Normal	0.9852	0.9449	0.9646
Accuracy		0.9831	
Macro avg.	0.9839	0.9702	0.9768
Weighted avg.	0.9832	0.9831	0.9830

To validate the optimization task of the approach, an experiment is conducted to select the best value of the XGB's max-depth hyper-parameter. We trained and tested the XGB using an exhaustive search algorithm with different values of max depth starting from 1 to 25. Figure 4 shows the confusion matrix of the XGB classifier on the test set with a max depth equal to 17 and default values of the other hyper-parameters.

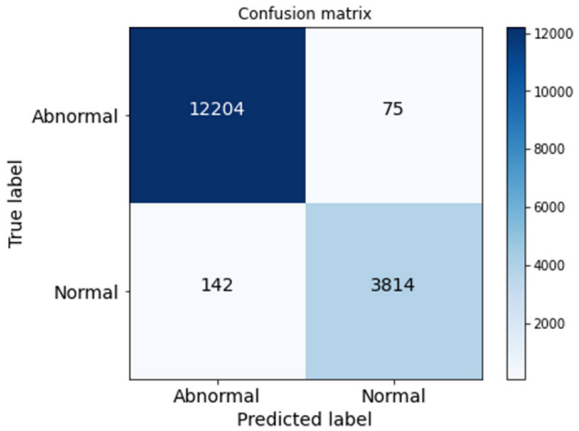

Fig. 4. Confusion matrix of XGB classifier on the test set using the best value of max-depth and default values of other hyper-parameters

From the results of the confusion matrix in Figure 4, we can see that the optimized XGB achieves 98.66% of accuracy and 0.9866 of weighted average precision, recall, and F1-measure. Also, we can notice that the optimized model attains 0.9846 of macro average precision, 0.9790 of macro average recall, and 0.9818 of macro average F1-measure as shown in Table 6.

The high results of the macro average of the optimized XGB classifier confirm that the proposed approach has a high performance to distinguish between abnormal and normal network traffic.

Table 6. The experimental results of proposed approach using the best value of max-depth and default values of XGB hyper-parameters

Class Label	Precision	Recall	F1-Measure
Abnormal	0.9885	0.9939	0.9912
Normal	0.9807	0.9641	0.9723
Accuracy		0.9866	
Macro avg.	0.9846	0.9790	0.9818
Weighted avg.	0.9866	0.9866	0.9866

To compare the performance results of optimized XGB classifier of the proposed approach with the Gradient-boosting decision tree (GBDT) classifier, Table 7 presents the experimental outcomes of GBDT in terms of precision, recall, accuracy and F1-measure.

Table 7. The experimental results of proposed approach using GBDT classifier

Class Label	Precision	Recall	F1-Measure
Abnormal	0.9708	0.9991	0.9847
Normal	0.9969	0.9067	0.9497
Accuracy		0.9766	
Macro avg.	0.9839	0.9529	0.9672
Weighted avg.	0.9772	0.9766	0.9762

The accuracy of the proposed approach is also compared with the accuracy result of some recent work developed for intrusion detection. Table 8 shows that the approach of our work achieves a high accuracy result (highlighted with black bold font) compared with the approaches and models proposed in [32–34] on the same dataset with the same partitioning ratio of the test set.

Table 8. Accuracy of the proposed approach compared with the accuracy result of some cited recent work

Research Work	Method/Approach/Model	Accuracy
Rajagopal et al. [32]	Two-class neural network	97%
Gudla et al. [33]	LSTMDL	94.11%
Kumar and Sharma [34]	MLP with Feature Selection	98.10%
This work	Optimized XGB with Feature Selection	**98.66%**

5 CONCLUSION AND FUTURE WORK

In this paper, an optimized approach using XGB classifier with correlation-based feature selection is developed to detect normal and abnormal traffic in intrusion systems. The developed approach optimized the max-depth parameter of XGB as a specified criterion to prune its trees and improve its performance significantly. The best value of the max-depth parameter is selected using an exhaustive search algorithm. The Pearson correlation coefficient-based method is used for feature selection and the attributes with more than or equal to 0.2 correlation coefficient with the target class label are selected as significant features. The approach is evaluated through a number of experiments conducted on the public UNSW-NB15 dataset that imitates the modern-day attacks of network traffic. The categorical attributes of the dataset are converted to numerical attributes using one-hot-encoded. In the evaluation experiments, the dataset is divided into two parts: 80% for testing and 20% for testing. The experimental results show the ability of the proposed approach to classify normal and abnormal traffic with high accuracy results (98.66%) compared with the current state-of-the-art work on the same dataset with the same partitioning ratio of the test set. In future work, the proposed approach will be applied on more than one dataset and will be used to classify the different types of attacks included in the abnormal network traffic inputs.

6 ACKNOWLEDGMENT

The authors would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq), Baghdad-Iraq, for its support in the present work. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444).

7 REFERENCES

[1] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," ACM Computing Surveys, vol. 47, no. 4, 2015. https://doi.org/10.1145/2716260

[2] M. Maabreh, I. Obeidat, E. A. Elsoud, A. Alnajjar and R. Alzyoud et al., "Towards data-driven network intrusion detection systems: Features dimensionality reduction and machine learning," International Journal of Interactive Mobile Technologies, vol. 16, no. 14, 2022. https://doi.org/10.3991/ijim.v16i14.30197

[3] A. Achmad, I. S. Areni, E. Palantei, M. S. Hadis and A. D. Achmad, "Smart electrical devices control with intrusion detection alert," International Journal of Interactive Mobile Technologies, vol. 16, no. 5, 2022. https://doi.org/10.3991/ijim.v16i05.26737

[4] Y. Y. Chung and N. Wahid, "A hybrid network intrusion detection system using simplified swarm optimization (sso)," Applied soft computing, vol. 12, no. 9, 2012. https://doi.org/10.1016/j.asoc.2012.04.020

[5] S. Ganapathy, K. Kulothungan, S. Muthurajkumar, M. Vijayalakshmi and P. Yogesh et al., "Intelligent feature selection and classification techniques for intrusion detection in networks: A survey," EURASIP Journal on Wireless Communications Networking, vol. 2013, no. 1, 2013. https://doi.org/10.1186/1687-1499-2013-271

[6] A. R. Muhsen, G. G. Jumaa, N. F. AL Bakri and A. T. Sadiq, "Feature selection strategy for network intrusion detection system (nids) using meerkat clan algorithm," International Journal of Interactive Mobile Technologies, vol. 15, no. 16, 2021. https://doi.org/10.3991/ijim.v15i16.24173

[7] A. A. Aburomman and M. B. I. Reaz, "Review of IDs development methods in machine learning," International Journal of Electrical Computer Engineering, vol. 6, no. 5, 2016. http://doi.org/10.11591/ijece.v6i5.pp2432-2436.
[8] E. Bahri, N. Harbi and H. N. Huu, "Approach based ensemble methods for better and faster intrusion detection," in Computational intelligence in security for information systems: Springer, 2011. https://doi.org/10.1007/978-3-642-21323-6_3
[9] A. A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," Computers Security Communication Networks, vol. 65, 2017. https://doi.org/10.1016/j.cose.2016.11.004
[10] R. Polikar, "Ensemble learning," in Ensemble machine learning: Springer, 2012. https://doi.org/10.1007/978-1-4419-9326-7_1
[11] R. Sikora, "A modified stacking ensemble machine learning algorithm using genetic algorithms," in Handbook of research on organizational transformations through big data analytics: IGi Global, 2015. https://doi.org/10.4018/978-1-4666-7272-7.ch004
[12] G. Zhao, Z. Shen, C. Miao and R. Gay, "Enhanced extreme learning machine with stacked generalization," in 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008. https://doi.org/10.1109/IJCNN.2008.4633951
[13] B. Zenko, L. Todorovski and S. Dzeroski, "A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods," in Proceedings 2001 IEEE International Conference on Data Mining, 2001. IEEE.
[14] H.-C. Yi, Z.-H. You, M.-N. Wang, Z.-H. Guo and Y.-B. Wang et al., "Rpi-se: A stacking ensemble learning framework for ncrna-protein interactions prediction using sequence information," BMC bioinformatics, vol. 21, no. 1, 2020. https://doi.org/10.1186/s12859-020-3406-0
[15] S. Diplaris, G. Tsoumakas, P. A. Mitkas and I. Vlahavas, "Protein classification with multiple algorithms," in Panhellenic conference on informatics, Springer, 2005. https://doi.org/10.1007/11573036_42
[16] N. C. Oza and K. Tumer, "Classifier ensembles: Select real-world applications," Information fusion, vol. 9, no. 1, 2008. https://doi.org/10.1016/j.inffus.2007.07.002
[17] S. Rajagopal, P. P. Kundapur and K. S. Hareesha, "A stacking ensemble for network intrusion detection using heterogeneous datasets," Security Communication Networks, vol. 2020, 2020. https://doi.org/10.1155/2020/4586875
[18] N. Chand, P. Mishra, C. R. Krishna, E. S. Pilli and M. C. Govil, "A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection," in 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Spring), IEEE, 2016. https://doi.org/10.1109/ICACCA.2016.7578859
[19] J. Vanerio and P. Casas, "Ensemble-learning approaches for network security and anomaly detection," in Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks, 2017. https://doi.org/10.1145/3098593.3098594
[20] X. Gao, C. Shan, C. Hu, Z. Niu and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," IEEE Access, vol. 7, 2019. https://doi.org/10.1109/ACCESS.2019.2923640
[21] W. Zhang, H. Ren, Q. Jiang and K. Zhang, "Exploring feature extraction and elm in malware detection for android devices," in International Symposium on Neural Networks, Springer, 2015. https://doi.org/10.1007/978-3-319-25393-0_54
[22] B. Kavitha, S. Karthikeyan and P. S. Maybell, "An ensemble design of intrusion detection system for handling uncertainty using neutrosophic logic classifier," Knowledge-Based Systems, vol. 28, 2012. https://doi.org/10.1016/j.knosys.2011.12.004
[23] J. Han, J. Pei and H. Tong, "Data mining: Concepts and techniques". Morgan Kaufmann, 2022.
[24] W.-H. Chen, S.-H. Hsu and H.-P. Shen, "Application of SVM and ANN for intrusion detection," Computers Operations Research, vol. 32, no. 10, 2005. https://doi.org/10.1016/j.cor.2004.03.019

[25] M. Govindarajan and R. Chandrasekaran, "Intrusion detection using neural based hybrid classification methods," Computer networks, vol. 55, no. 8, 2011. https://doi.org/10.1016/j.comnet.2010.12.008

[26] A. A. Aburomman and M. B. I. Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," Applied Soft Computing, vol. 38, 2016. https://doi.org/10.1016/j.asoc.2015.10.011

[27] R. R. Karthick, V. P. Hattiwale and B. Ravindran, "Adaptive network intrusion detection system using a hybrid approach," in 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012), IEEE, 2012.

[28] L. Li, Y. Yu, S. Bai, Y. Hou and X. Chen, "An effective two-step intrusion detection approach based on binary classification and k-NN," IEEE Access, vol. 6, 2017. https://doi.org/10.1109/ACCESS.2017.2787719

[29] A. J. Malik, W. Shahzad and F. A. Khan, "Binary PSO and random forests algorithm for probe attacks detection in a network," in 2011 IEEE Congress of Evolutionary Computation (CEC), IEEE, 2011. https://doi.org/10.1109/CEC.2011.5949682

[30] Y. Zhou, G. Cheng, S. Jiang and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Computer Networks, vol. 174, 2020. https://doi.org/10.1016/j.comnet.2020.107247

[31] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," Computers Security Communication Networks, vol. 70, 2017. https://doi.org/10.1016/j.cose.2017.06.005

[32] S. Rajagopal, K. S. Hareesha, and P. P. Kundapur, "Feature relevance analysis and feature reduction of UNSW NB-15 using neural networks on MAMLS," In *Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2018*, vol. 1, pp. 321–332. Springer Singapore, 2020. https://doi.org/10.1007/978-981-15-1081-6_27

[33] S. P. K. Gudla, S. K. Bhoi, S. R. Nayak, K. K. Singh, A. Verma and I. Izonin, "A deep intelligent attack detection framework for fog-based IoT systems," Computational Intelligence and Neuroscience, 2022. https://doi.org/10.1155/2022/6967938

[34] A. Kumar and D. Sharma, "Intrusion detection system using multi-layer perceptron with grid search cv," International Journal for Modern Trends in Science and Technology, vol. 8, no. 07, 2022. https://doi.org/10.46501/IJMTST0807016

[35] V. F. Rodriguez-Galiano, J. A. Luque-Espinar, M. Chica-Olmo and M. P. Mendes, "Feature selection approaches for predictive modelling of groundwater nitrate pollution: An evaluation of filters, embedded and wrapper methods," Science of the Total Environment, vol. 624, 661–672, 2018. https://doi.org/10.1016/j.scitotenv.2017.12.152

[36] I. M. Nasir, M. A. Khan, M. Yasmin, J. H. Shah and M. Gabryel et al., "Pearson correlation-based feature selection for document classification using balanced training," Sensors, vol. 20, no. 23, 2020. https://doi.org/10.3390/s20236793

[37] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang and I. Cohen, "Pearson correlation coefficient," Noise Reduction in Speech Processing, 1–4, 2009. https://doi.org/10.1007/978-3-642-00296-0_5

[38] T. Chen, T. He, M. Benesty, V. Khotilovich and Y. Tang et al., "Xgboost: Extreme gradient boosting," R package version 0.4-2, vol. 1, no. 4, 2015.

[39] S. González, S. García, J. D. Ser, L. Rokach, and F. Herrera, "A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities," Information Fusion, vol. 64, 205–237, 2020. https://doi.org/10.1016/j.inffus.2020.07.007

[40] C. Bentéjac, A. Csörgő and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," Artificial Intelligence Review, vol. 54, no. 3, 2021. https://doi.org/10.1007/s10462-020-09896-5

[41] J. Bergstra, R. Bardenet, Y. Bengio and B. Kégl, "Algorithms for hyper-parameter optimization," Advances in Neural Information Processing Systems, vol. 24, 2011.

[42] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in 2015 military communications and information systems conference (MilCIS), 2015, IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

[43] N. Moustafa and 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," Information Security Journal: A Global Perspective, vol. 25, no. 1–3, 2016. https://doi.org/10.1080/19393555.2015.1125974

8 AUTHORS

Ghassan Muslim Hassan is a Ph.D. who works in the Computer Science Department of Al-Mustansiriyah University. He received his B.Sc. degree in Electrical and Communication Engineering (1985) from the University of Technology/Iraq, and M.Sc. degree in Computer Science (1991) from the University of Technology/Iraq. He obtained his Ph.D. in Computer Science in the field of Communication Technology and Networks from The National University of Malaysia (Universiti Kebangsaan Malaysia, UKM) in 2019. He joined Al-Mustansiriyah University as a Lecturer in 2003. He has more than 25 years of academic experience in Iraqi and Libyan Universities. He taught many courses based on Network, Communication, Network Security, Data Structure, Logic Design, and more. He has published several peer-reviewed articles in high-impact journals. His research involves Wireless Communication, Speech Recognition, Internet of Things (IoT), and Data Structure (Email: gmhalsaddi@uomustansiriyah.edu.iq).

Abdu Gumaei received his Ph.D. degree in Computer Science from King Saud University in 2019. He is currently an Assistant Professor at Computer Science Department, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Kharj, Saudi Arabia. He worked as a lecturer and taught many courses such as programming languages in the computer science department, Taiz University, Yemen. He has authored more than 120 journal and conference papers in well-reputed international journals. He has received a patent from the United States Patent and Trademark Office (USPTO) in 2013. His main areas of interest are software engineering, image processing, computer vision, machine learning, networks, and the Internet of Things (IoT) (Email: a.gumaei@psau.edu.sa).

Abed Alanazi received his Ph.D. degree in Computer Science and Engineering, University of South Florida, USA Master's degree in Computer Science, Florida Institute of Technology, USA and Bachelor's degree in Computer, King Saud University, Saudi Arabia. He is currently an Assistant Professor at Computer Science Department, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Kharj, Saudi Arabia. His research interests are Cyber Security, IoT Security Issues, Artificial Intelligence, and Cloud Computing. He is a head of the Department of Computer Science, College of Computer Science and Engineering (Email: ad.alanazi@psau.edu.sa).

Samah M. Alzanin received the bachelor's degree in computer and mathematics from Ibb University, Ibb, Yemen. The master and Ph.D. degrees in computer science from King Saud University, Riyadh, Saudi Arabia. She is currently an Assistant Professor with Applied College, Prince Sattam bin Abdulaziz University, Kharj, Saudi Arabia. Her research interests NLP, machine learning, and AI applications (Email: s.alzanin@psau.edu.sa).