Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
() : –
© TÜBİTAK
doi:

Network intrusion detection based on machine learning strategies: performance comparisons on imbalanced wired, wireless, and software-defined networking (SDN) network traffics

Hilal HACILAR[1,**], **Zafer AYDIN**[1], **Vehbi Çağrı Güngör**[2]
[1]Department of Computer Engineering, Faculty of Engineering, Abdullah Gül University, Kayseri, Turkiye
[2]Turkcell, İstanbul, Turkiye

Received: 30.10.2023 • **Accepted/Published Online:** 24.04.2024 • **Final Version:** ..202

Abstract: The rapid growth of computer networks emphasizes the urgency of addressing security issues. Organizations rely on network intrusion detection systems (NIDSs) to protect sensitive data from unauthorized access and theft. These systems analyze network traffic to detect suspicious activities, such as attempted breaches or cyberattacks. However, existing studies lack a thorough assessment of class imbalances and classification performance for different types of network intrusions: wired, wireless, and software-defined networking (SDN). This research aims to fill this gap by examining these networks' imbalances, feature selection, and binary classification to enhance intrusion detection system efficiency. Various techniques such as SMOTE, ROS, ADASYN, and SMOTETomek are used to handle imbalanced datasets. Additionally, eXtreme Gradient Boosting (XGBoost) identifies key features, and an autoencoder (AE) assists in feature extraction for the classification task. The study evaluates datasets such as AWID, UNSW, and InSDN, yielding the best results with different numbers of selected features. Bayesian optimization fine-tunes parameters, and diverse machine learning algorithms (SVM, kNN, XGBoost, random forest, ensemble classifiers, and autoencoders) are employed. The optimal results, considering F1-measure, overall accuracy, detection rate, and false alarm rate, have been achieved for the UNSW-NB15, preprocessed AWID, and InSDN datasets, with values of [0.9356, 0.9289, 0.9328, 0.07597], [0.997, 0.9995, 0.9999, 0.0171], and [0.9998, 0.9996, 0.9998, 0.0012], respectively. These findings demonstrate that combining Bayesian optimization with oversampling techniques significantly enhances classification performance across wired, wireless, and SDN networks when compared to previous research conducted on these datasets.

Key words: Network intrusion detection systems (NIDS), network anomaly detection, deep learning, Bayesian optimization, class imbalance, software-defined networking (SDN)

1. Introduction

The exponential growth of computer networks has led to a significant rise in security concerns, particularly regarding network intrusions. The role of network intrusion detection systems (NIDSs) is of remarkable importance in protecting organizations' IT infrastructure from potential cyber attacks. As security incidents and financial losses per incident continue to increase, the detection of network attacks has become a top priority and a significant challenge for researchers in computer science and network security. The primary objective of NIDS is to develop predictive models capable of distinguishing normal network activities from abnormal ones. Various studies in the literature have explored predictive modeling techniques, including XGBoost, artificial neural

*Correspondence: hilal.hacilar@agu.edu.tr

This work is licensed under a Creative Commons Attribution 4.0 International License.

networks (ANN), deep learning, and other conventional machine learning algorithms. However, class imbalance problems often arise in these predictive modeling studies [1]. Many approaches, such as SMOTE (synthetic minority oversampling technique) [2], have been suggested as potential solutions to tackle this issue and have shown their efficacy by improving classification accuracy. In conjunction with the issue of class imbalance, an effective network anomaly detection system must also address the challenges posed by various attack patterns, encrypted data transmissions, and the need for real-time application performance. Researchers have conducted studies on various techniques for selecting features and integrating classifiers with network intrusion detection systems in order to improve the ability to predict network attacks. Nevertheless, the lack of a widely accepted approach for detecting network intrusions is obvious, and a noticeable lack of scientific studies comparing the different feature selection and classification algorithms performances with respect to distinct evaluation metrics is apparent. In order to address the existing research gap, the primary aim of this study is to conduct an analysis of network intrusions inside wired, wireless, and software-defined networking (SDN) settings. This analysis will specifically focus on the challenges associated with classification and class imbalance. This study utilizes many techniques such as class imbalance strategies, feature selection, hyperparameter optimization, and classification methods in order to accurately detect network intrusions. In this context, this study employs various methods, including ROS, SMOTE, SMOTETomek, and ADASYN (adaptive synthetic sampling) [3], to address the issue of unbalanced datasets. Additionally, XGBoost has been utilized to assess the importance of features during the feature selection process. The main contributions of this study to the literature are listed below:

(i) To alleviate the computational workload and minimize training time, XGBoost feature selection has been applied to pinpoint the most informative features within IDS datasets.

(ii) An efficient hyperparameter optimization technique called Bayesian optimization has been employed to fine-tune the model's parameters. This optimization process aims to identify the best combination of hyperparameters that yield optimal performance while minimizing the required training time.

(iii) By applying imbalanced strategies including SMOTE, SMOTETomek, and ADASYN, the model can better handle the challenges posed by imbalanced datasets and improve its performance in detecting intrusions accurately while minimizing the impact of class imbalance.

(iv) Various ML models have been constructed to classify intrusions and normal flows, followed by a comprehensive evaluation using various metrics such as F1-measure, overall accuracy, false alarm rate, and detection rate (Figure 1 provides an overview of these steps). It has been observed that the proposed classification methodologies produce superior results compared to the existing literature (Table 1).

(v) To the best of our knowledge, no study has been conducted to evaluate different network intrusion datasets, such as wired, wireless, and SDN, together, considering class imbalance, feature selection, and hyperparameter optimization tasks. This study aims to address this gap.

The organization of the paper is as follows: Section 2 examines related work on network intrusion detection. Section 3 introduces the datasets, providing detailed information about their network features and preprocessing steps. It then describes the approach for feature selection and extraction using XGBoost and autoencoders. Section 4 presents the evaluation metrics used in classification conducted on the UNSW-NB15 [4], AWID [5], and InSDN [6] datasets. Section 5 provides the conclusion of this study.

2. Related work

The literature on network anomaly detection suggests a wide amount of algorithms, including both deep learning techniques and standard machine learning algorithms. Researchers have emphasized the importance of rapid

response in network intrusion detection systems. For instance, Potluri and Diedrich [7] have utilized parallel computing and deep learning algorithms on the NSL-KDD dataset, achieving an accuracy score of 97.5%. Hoang [8] has employed parallel genetic programming on the AWID dataset, reducing computing costs and achieving precision, recall, and F1-measure values of 0.785, 0.78, and 0.78, respectively. Kolukisa et al. [9] also have trained logistic regression parameters via parallel artificial bee colony algorithm and achieved significant results on UNSW_NB15 dataset.

Tree-based machine learning algorithms, such as XGBoost, have demonstrated effectiveness and feasibility in network intrusion detection systems [10–13]. Kevric et al. [10] utilized NB trees and random trees as a hybrid technique on the NSL-KDD dataset, achieving an accuracy of 89.24%. Pattawaro and Polprasert [11] employed an ensemble approach combining kNN and XGBoost on the NSL-KDD dataset, achieving an accuracy of 84.4%. Dhaliwal et al. [13] performed XGBoost on the NSL-KDD dataset, attaining values of 98.70% accuracy and 98.76% F1-score. In the study of Logeswari et al. [14], a novel hybrid feature selection and LightGBM-based Intrusion Detection System (IDS) was proposed for SDN. They tested their proposed model on the NSL-KDD dataset and achieved notable results.

Prior to classification, several research studies have employed dimension reduction techniques such as linear discriminant analysis (LDA), principal component analysis (PCA), and autoencoder (AE). LDA and PCA are linear transformation methods, with PCA being unsupervised and LDA being supervised. Autoencoders, on the other hand, are neural networks that attempt to compress input data into a smaller model using an encoder and a bottleneck. Autoencoders operate in a nonlinear manner, similar to PCA but with greater flexibility. Research studies have generated autoencoder algorithms for dimension reduction and classification tasks, achieving high evaluation scores [15–21].

Class imbalance is a prevalent issue in machine learning, particularly in fields such as medical diagnosis, fraud detection, and anomaly detection, where the number of instances of one class significantly outnumbers the instances of the other. This imbalance can lead to biased models that perform well on the majority class but poorly on the minority class, thus compromising the overall predictive performance. Various techniques, such as resampling methods, cost-sensitive learning, and advanced algorithms like SMOTE, are employed to address this challenge and improve model robustness. Wheelus et al. [22] tackle the issue mentioned and demonstrate that SMOTE proves to be more effective than alternative algorithms in addressing class imbalance, particularly in terms of ROC area compared to specific machine learning algorithms. Abdulhammed et al. [23] acknowledge the challenge of class imbalance and implement preprocessing techniques designed for imbalanced datasets, achieving a remarkable 99.99% accuracy in the CIDDS-001 intrusion dataset. Additionally, Ran et al. [24] adopt a semisupervised learning approach as a substitute for traditional supervised machine learning algorithms, incorporating undersampling to effectively address the class imbalance problem. Abdelkhalek and Mashaly [25] achieve significant results by combining the ADASYN and Tomek links sampling techniques.

Other deep learning methods, including deep belief networks (DBN), deep neural networks (DNN), convolutional neural network (CNN), recurrent neural network (RNN) and LSTM-based autoencoder, have been utilized to construct predictive models. These approaches have demonstrated high accuracy rates on datasets such as KDD Cup '99, NSL-KDD, and InSDN [21, 22, 26, 27]. Some studies achieve significant results by employing deep learning algorithms in a hybrid manner. For instance, Qazi et al. [28] apply CNN and RNN algorithms in a hybrid manner to the CICIDS-2018 dataset, resulting in notable outcomes. Regarding software-defined networking (SDN), research studies have highlighted the importance of detecting distributed denial of service (DDoS) attacks in SDN networks. Bhayo et al. [29] proposed machine learning-based approach

to detect DDoS attacks in an SDN-WISE IoT controller. They integrated naive Bayes (NB), decision tree (DT), and support vector machine (SVM)-based detection module into the controller and achieved a high level of accuracy.

Various models, including decision trees, naive Bayes, k-nearest neighbors, and extra trees, have been applied with majority voting to defend against DDoS attacks [30]. Yoo et al. [31] proposed a hybrid model that combines a random forest and a deep learning model to classify files as either malware or benign. By employing hybrid majority voting rules, they achieved a significant improvement in detection rate. LSTM-based autoencoder and one-class SVM methods have been used for encoding and classification in SDN networks, achieving high accuracy rates [32]. Ensemble approaches based on k-means++ and random forest have also demonstrated excellent precision and recall in detecting attacks on SDN datasets [33]. Transforming SDN network traffic tabular data into image data and employing modified CNN models have also yielded high accuracy rates [34]. Another studies [35, 36] use the widely used and benchmark UNSW-NB15 dataset, achieving significant results with DBN and SVM, respectively.

While all the methods mentioned make valuable contributions to the existing literature, to the best of our knowledge, there is no study in the literature that systematically assesses and compares anomaly detection across datasets generated in diverse network environments, considering factors such as class imbalance, hyperparameter optimization, and feature selection. To address these challenges in intrusion detection systems (IDS), this study evaluates wired, wireless, and SDN networks in terms of feature selection, class imbalance problems, hyperparameter optimization, and binary classification tasks. By considering these factors, the goal is to efficiently detect network anomalies in different network environments. Table 1 shows that proposed methodologies yield superior classification outcomes compared to the literature.

Table 1. Comparison of existing works using the UNSW-NB15, AWID, and InSDN datasets.

Ref	Dataset	Method	Precision	Recall	F1	Acc	Year
[21]	UNSW-NB15	LDA and random tree	0.861	0.865	-	86.46%	2018
[35]	UNSW-NB15	Improved DBN	-	-	-	86.49%	2020
[36]	UNSW-NB15	SVM	-	-	-	85.99%	2019
[9]	UNSW-NB15	ABC-LR	-	-	0.8826	88.25%	2023
This study	**UNSW-NB15**	**Random forest (with Bayesian opt.)**	**0.93**	**0.94**	**0.9356**	**92.89%**	**-**
[12]	AWID	Random forest	0.96	0.96	0.95	99.106%	2018
[8]	AWID	GP	0.79	0.78	0.78	-	2018
[8]	AWID	Karoo-GP	0.82	0.79	0.80	-	2017
[24]	AWID	DNN based on ladder network	-	-	-	99.28%	2019
[37]	AWID	Hunger Games Search and Remora Optimization	0.9976	0.994	0.9958	99.16%	2022
This study	**AWID**	**Random forest (with Bayesian opt.)**	**0.99**	**0.99**	**0.997**	**99.95%**	**-**
[30]	InSDN	V-NKDE	0.998	0.998	0.998	99.84%	2021
[32]	InSDN	LSTM-autoencoder-OC-SVM	0.93	0.93	0.93	90.50%	2020
This study	**InSDN**	**AE + random forest (with Bayesian opt.)**	**0.99**	**0.99**	**0.9998**	**99.96%**	**-**

Acc = accuracy, Ref = reference, F1 = F1-score.

3. Materials and methods

In this research, three publicly available datasets are used: AWID, UNSW NB15, and InSDN. The AWID dataset focuses on wireless network attacks and contains 155 features. The training set has 1,795,575 instances, and the testing set has 575,643 instances. The UNSW NB15 dataset includes wired network attacks, and it contains various attack categories along with normal traffic. The number of samples in the training set is 175,341, and the testing set is 82,332. The InSDN dataset consists of SDN traffic data and includes attacks targeting different layers and SDN-specific attacks. It contains 77 features and 343,939 instances of both normal and attack traffic. Before conducting classification experiments, preprocessing steps have been performed on the AWID dataset to transform it into an efficient format. The UNSW NB15 dataset does not require preprocessing. In the InSDN dataset, low-variance features have been removed. To address the class imbalance problem, oversampling techniques such as SMOTE, SMOTETomek, and ADASYN have been applied. Feature selection has been carried out using XGBoost to identify the most relevant network traffic features. Bayesian optimization has been utilized to optimize the hyperparameters of existing machine learning algorithms as well as the proposed technique. Finally, evaluation metrics have been computed to assess the performance of the classification methods employed in the study.

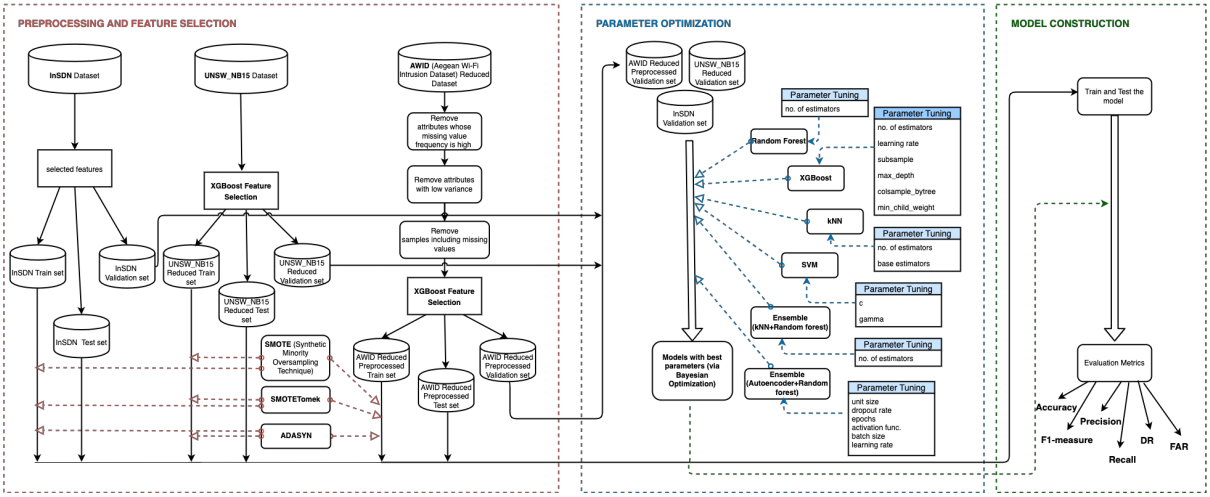

Figure 1. A schematic representation of our methodology: preprocessing, parameter optimization, and model construction.

3.1. Dataset and data preprocessing

Data preprocessing constitutes a crucial and fundamental step in data mining, and it encompasses converting raw data into a format that is efficient and relevant for analysis. In the context of this study, data preprocessing techniques have been applied to the AWID, InSDN and UNSW-NB15 datasets.

The AWID dataset contains 155 attributes; however, not all of them play a role in training the model. Within the AWID dataset, data varies in value and type—ranging from discrete and continuous to symbolic—creating a wide-ranging value spectrum. These diverse data characteristics pose a challenge for classifiers to accurately understand the underlying details. Hence, an essential step in classification is the preprocessing phase to navigate this complexity. In the literature, several researchers ([8, 12, 24, 37]) mentioned in the comparison table (Table 1) have employed a reduced version of the AWID dataset and applied different preprocessing strategies. First, specific features that do not affect variance and have a high number of missing values have

been excluded from all studies. However, the exclusion of features varies from one study to another based on their specific criteria and the methods used for feature selection. Secondly, they have applied different strategies to samples containing missing values. Vaca et al. [12] have removed missing values with the most occurred values. Ran et al. [24] have replaced missing values with zeros. Kumar et al. [37] also used replacing missing values. In this study, the AWID dataset has been imported into SPSS [1], and specific features with low variance have been excluded based on their feature frequency specifications. Features with a large number of missing values have been removed from the dataset. Replacing or deleting missing values has both advantages and drawbacks. Given that only 2% of the original data contains missing values and to avoid errors or biases that may arise from "replacing" missing values, "deleting" was preferred in this study. Consequently, any remaining samples with missing values have been eliminated. As a result of these preprocessing methods, 20 out of the original 155 features from the AWID dataset have been retained.

Similarly, irrelevant features, such as destination and source IP addresses, flow IDs, and timestamps, were removed from the InSDN dataset during the preprocessing step. These attributes were eliminated as they were deemed irrelevant for the classification task. With the application of these preprocessing techniques, the datasets were cleansed and rendered suitable for subsequent classification and analysis.

The UNSW-NB15 dataset offers distinct training and testing datasets. The training set comprises 175,341 samples, with 56,000 labeled as "normal" and 119,341 labeled as "abnormal." Similarly, the testing set includes 82,332 samples, with 37,000 labeled as "normal" and the remaining 45,332 labeled as "abnormal" traffic samples. Hence, with the UNSW-NB15 dataset containing categorical features, the one-hot encoding technique is employed to convert these categorical features into numeric values. After encoding, the three categorical features, including 'service,' 'state,' and 'proto' in the UNSW-NB15 dataset, contribute to an increased from 45 to 196 features.

3.2. Evaluation metrics

A critical stage in model development, the evaluation of machine learning algorithms involves assessing the model's performance using various evaluation metrics, which obtain from confusion matrix (Table 2). Although accuracy (1) is commonly used, it is essential to consider additional metrics for a comprehensive evaluation of the model's performance. When faced with imbalanced datasets, such as intrusion detection and others with a greater proportion of normal samples than abnormal samples, accuracy may fail to provide a comprehensive understanding. Metrics such as precision, F1-score (2), recall, false alarm rate (FAR) (3), and detection rate (DR) (4) become critically important in such situations. The metric of precision assesses the model's capability to accurately detect positive instances by diving the number of correctly predicted positive samples by the number of samples predicted as positive. Recall, also known as sensitivity or true positive rate (TPR), measures the percentage of real positive samples that are correctly identified. The F1-score, a combination of precision and recall, provides an integrated performance evaluation that considers both aspects. The detection rate is the ratio of true positive samples to the actual positive samples. The false alarm rate is calculated by dividing the number of false positive samples by the overall count of actual negative samples, indicating the model's tendency to incorrectly label negative instances as positive. In addition to accuracy, in this study, other evaluation metrics such as precision, F1-score, recall, detection rate, and false alarm rate have been used to assess the performance of an algorithm for machine learning in a more comprehensive manner.

[1]IBM Corp. IBM SPSS Statistics [online]. Website https://www.ibm.com/support/pages/downloading-ibm-spss-statistics-25 [accessed 01 Semptember 2022].

Table 2. Confusion matrix.

	Predicted negative	**Predicted positive**
Actual negative	TN	FP
Actual positive	FN	TP

$$Accuracy = \frac{TP + TN}{TN + FP + FN + TP} \tag{1}$$

$$F1 - \ score = \frac{2 * TP}{2 * TP + FP + FN} \tag{2}$$

$$False\ alarm\ rate\ (FAR) = \frac{FP}{TN + FP} \tag{3}$$

$$Detection\ rate\ (DR\ or\ TPR) = \frac{TP}{FN + TP} \tag{4}$$

3.3. Class imbalance problem

When the minority class in a dataset contains fewer samples than the majority class, traditional machine learning algorithms may struggle to perform well. This study employs oversampling rather than under sampling to resolve this issue and prevent the loss of informative minority class samples.

The SMOTE method has been applied as the initial oversampling technique on the preprocessed datasets. SMOTE is commonly used to fix class imbalance by creating synthetic data points for the minority class, using nearest neighbors. This balances class distribution and boosts classification accuracy and F1-measure.

In addition to SMOTE, the SMOTETomek algorithm has been applied to all datasets. SMOTETomek combines the capabilities of SMOTE for generating synthetic data with the Tomek link under sampling method. Tomek links, which are pairings of samples belonging to distinct classes that are in close range to one another, can be eliminated to enhance the differentiation between classes.

Furthermore, the ADASYN algorithm has been implemented as the final step of oversampling. ADASYN is an enhanced version of SMOTE that adds small random values to the synthetic samples generated by SMOTE. This improves the performance of the classification model by increasing the diversity and realism of the synthetic samples.

By using oversampling methods, such as SMOTE, SMOTETomek, and ADASYN, the problem of imbalanced class distribution can be reduced. These algorithms help the classification models learn better from the minority class samples, leading to improved accuracy and F1-measure for classification tasks.

3.4. Feature selection via eXtreme Gradient Boosting (XGBoost) algorithm

In this study, the XGBoost algorithm has been applied for feature selection in the network intrusion detection model construction. Feature selection plays a crucial role in reducing computation costs and improving the classification performance of the model.

XGBoost assigns relative relevance values to features based on their importance in making critical decisions on decision trees. The more a feature is used in decision making, the higher its importance score

becomes. By applying a threshold, which has been determined based on the point where feature importance scores decline rapidly, the top-ranked features can be selected.

Figures 2 and 3 provide visual representations of the importance scores of each feature for the AWID and the UNSW-NB15 datasets. These figures present the relative importance scores of the selected features based on their respective weights, where the cumulative importance scores of all features amount to one. For the AWID dataset, experiments encompassed 20 features ranging from "wlan.fc.subtype" to "radiotap.channel.type.cck," as well as 12 features covering "wlan.fc.subtype" to "wlan.fc.frag," as indicated in Table 3. Similarly, experiments were conducted for the UNSW-NB15 dataset, encompassing 37 features spanning from "sttl" to "service=ftp," and 20 features from "sttl" to "dloss," as indicated in Table 4. Feature selection has not been applied to the InSDN dataset because it produces efficient results in its current form ([6]).

The feature selection step helps to identify the most relevant and informative features, which can lead to improved classification performance. By selecting critical attributes, the model can efficiently reduce computational costs while simultaneously improving its accuracy in identifying network intrusions.

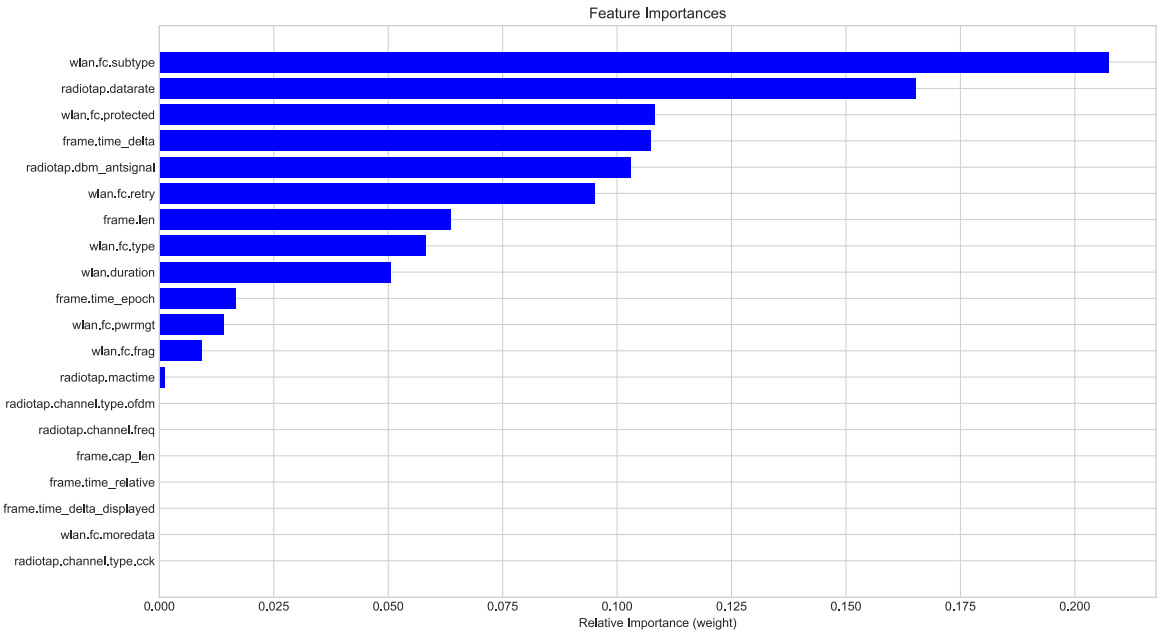

Figure 2. The top 20 features with higher relative importance in the AWID dataset. The Y axis corresponds to the names of these features, and the X axis corresponds to the relative importance values of the corresponding features.

3.5. Feature extraction via autoencoder

Figure 4 depicts the autoencoder (AE), an unsupervised deep learning model designed to handle high-dimensional data. The AE compresses the input data into a bottleneck hidden layer, representing the encoded input data. The decoder component of the AE reconstructs the original input data by leveraging the encoded data and minimizing the reconstruction loss. During the training process, AEs aim to minimize the reconstruction error. In this particular context, the AE model has been trained using the training set and validated using the vali-

dation set. Following the Bayesian optimization process, a random forest classifier has been applied to the best encoded samples (compressed data shown in Figure 4) obtained from the AE model.

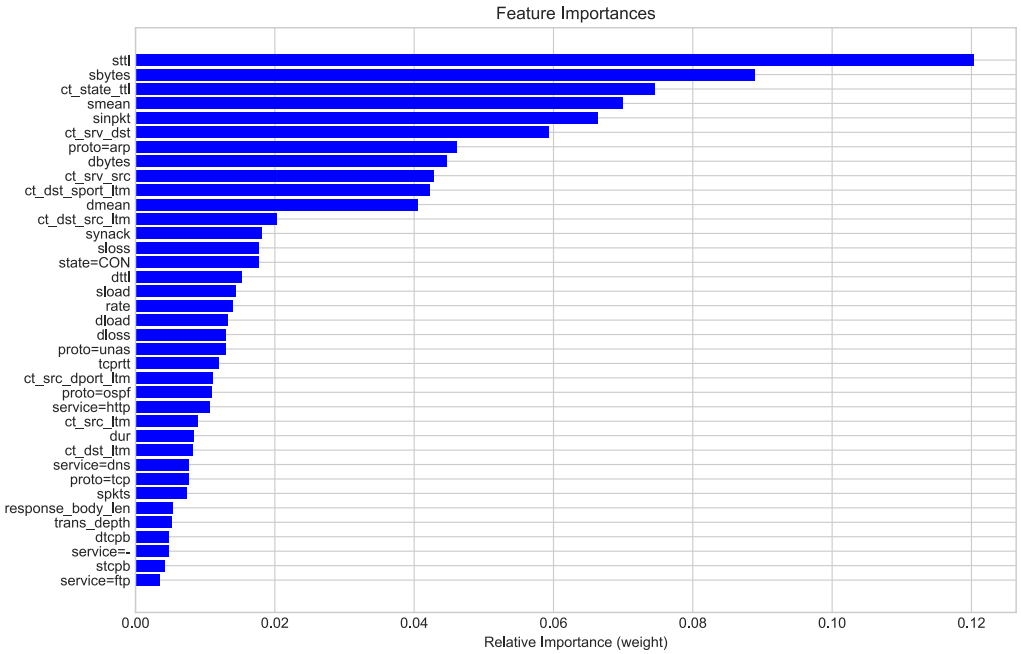

Figure 3. The top 37 features with high relative importance in the UNSW-NB15 dataset. The Y axis corresponds to the names of these features, and the X axis corresponds to the relative importance values of the corresponding features.

3.6. Hyperparameter optimization using Bayesian optimization

Optimizing parameters is a critical aspect of learning algorithms. While machine learning algorithms come with default parameters, tuning them is essential to achieve optimal performance. In this study, we explore various classification algorithms with different hyperparameters. To prevent overfitting, we employ validation sets instead of test sets. Stratified random sampling has been utilized to create balanced validation sets, which involve randomly sampling 30% of the imbalanced versions of the AWID and UNSW-NB15 training sets.

Table 5 presents the tuned hyperparameters for each classification algorithm along with the search approaches employed. For an SVM model, "c" and "gamma" are hyperparameters that require careful balancing between bias and variance. To find optimal values for these parameters, a randomized search strategy has been employed. The parameter named "n_estimators" determines the total number of trees present in the forest when employing algorithms such as random forest. In the XGBoost model, "learning_rate" is a hyperparameter that determines the weight adjustment of newly added trees. Furthermore, the parameters "max_depth", "min_child_weight", "colsample_bytree", and "subsample" correspond to the following properties in the context of base learners: the maximum depth allowed for each tree, the ratio of columns to consider when building each tree, the minimum required sum of instance weight in a child node, and the ratio of training instances to use during the construction of individual trees, respectively.

Table 3. Selected features and their descriptions of the AWID dataset.

Selected features	**Descriptions**
frame.time_epoch	Epoch time
frame.time_delta	Time delta from previous captured frame
frame.len	Frame length on the wire
radiotap.datarate	Data rate (Mb/s)
radiotap.dbm_antsignal	Antenna signal
wlan.fc.type	Types of 802.11 packets
wlan.fc.subtype	Spesific types of 802.11 packets
wlan.fc.frag	Control bit for more fragments coming or not
wlan.fc.retry	Control bit for frame is a retransmission or not
wlan.fc.pwrmgt	Control bit for station will stay awake or sleep
wlan.fc.protected	Control bit for MSDU payload encrypted or not
wlan.duration	Duration
frame.time_delta_displayed	Time delta from previous displayed frame
frame.time_relative	Time since reference or first frame
frame.cap_len	Frame length stored into the capture file
radiotap.mactime	MAC timestamp
radiotap.channel.freq	Channel frequency
radiotap.channel.type.cck	Complementary code keying (CCK)
radiotap.channel.type.ofdm	Orthogonal frequency-division multiplexing (OFDM)
wlan.fc.moredata	Frame control field more data

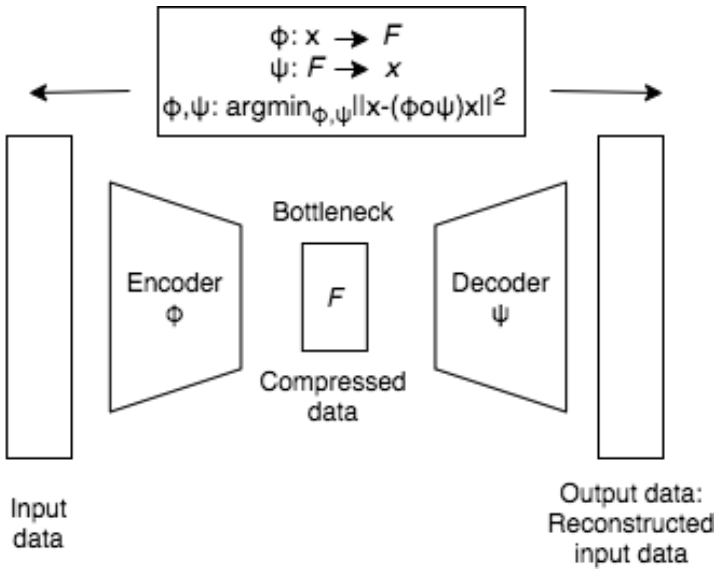

Figure 4. Illustration of the autoencoder model.

4. Experiments and results

The research methodology for this study comprises three primary phases. First, the contribution of feature selection using XGBoost has been evaluated in the classification problem. Second, the impact of different imbalance strategies on classification outcomes has been investigated. Third, the effectiveness of Bayesian hyper parameter optimization in classification has been assessed. Performance metrics have been compared across wired, wireless, and SDN networking flows.

Table 4. Selected features and their descriptions of the UNSW-NB15 dataset.

Selected features	**Descriptions**
dur	Record total duration
proto=tcp	Transaction protocol
proto=arp	Transaction protocol
proto=ospf	Transaction protocol
proto=unas	Transaction protocol
service=-	http, ftp, ssh, dns .., else (–)
service=http	http, ftp, ssh, dns .., else (–)
service=dns	http, ftp, ssh, dns .., else (–)
service=ftp	http, ftp, ssh, dns .., else (–)
state=CON	The state and its dependent protocol, e.g., ACC, CLO, else (–)
spkts	Source to destination packet count
sbytes	Source to destination bytes
dbytes	Destination to source bytes
rate	there is no description
sttl	Source to destination time to live
dttl	Destination to source time to live
sload	Source bits per second
dload	Destination bits per second
sloss	Source packets retransmitted or dropped
dloss	Destination packets retransmitted or dropped
sinpkt	Source inter-packet arrival time (mSec)
dtcpb	Destination TCP base sequence number
stcpb	Source TCP base sequence number
tcprtt	The sum of 'synack' and 'ackdat' of the TCP.
synack	The time between the SYN and the SYN_ACK packets of the TCP.
smean	Mean of the flow packet size transmitted by the src
dmean	Mean of the flow packet size transmitted by the dst
trans_depth	The depth into the connection of http request/response transaction
response_body_len	The content size of the data transferred from the server's http service.
ct_srv_src	No. of connections that contain the same service and source address in 100 connections according to the last time.
ct_state_ttl	No. for each state according to specific range of values for source/destination time to live.
ct_dst_ltm	No. of connections of the same destination address in 100 connections according to the last time.
ct_src_dport_ltm	No. of connections of the same source address and the destination portin 100 connections according to the last time.
ct_dst_sport_ltm	No. of connections of the same destination address and the source port in 100 connections according to the last time.
ct_dst_src_ltm	No. of connections of the same source and the destination address in 100 connections according to the last time.
ct_src_ltm	No. of connections of the same source address in 100 connections according to the last time.
ct_srv_dst	No. of connections that contain the same service and destination address in 100 connections according to the last time .

For the classification task, several machine learning algorithms have been employed. The SVM algorithm has been used to evaluate IDS datasets with respect to the linearity aspect. Tree-based models, including random forest and XGBoost, have been utilized, which make use of if-then rules for classification. The k-nearest neighbor algorithm has been applied both independently and in combination with random forests to create an ensemble classifier. These classification methods have been implemented using the scikit-learn library in Python.

Table 5. Ranges of classifier's hyperparameters used for Bayesian optimization.

Method	**Parameter**	**Lowest**	**Highest**
Random forest	n_estimators	10	500
Random forest	min_samples_leaf	1	5
Random forest	max_depth	1	150
Random forest	min_samples_split	2	10
SVM	c	0.001	1
SVM	gamma	0.01	1
XGBoost	n_estimators	10	500
XGBoost	subsample	0.3	1
XGBoost	max_depth	2	10
XGBoost	colsample_bytree	0.1	0.6
XGBoost	learning_rate	0.01	0.07
XGBoost	min_child_weight	1	5
Autoencoder	learning rate	10^{-8}	10^{-1}
Autoencoder	no of hidden units	5	50
Autoencoder	dropout rate	0	0.5
Autoencoder	batch size	1	1024
Autoencoder	no of epochs	1	50

When evaluating the performance of classification models using accuracy, it is important to address issues related to imbalanced datasets. The number of samples from different classes should not vary significantly. Additionally, in this study, other evaluation metrics such as F1 score, precision, recall, detection rate, and false alarm rate have been utilized.

Tables 6–8 provide a summary of the performance results from the experiments with optimal hyperparameters obtained through Bayesian optimization. These experiments include ablation studies, that is, the effect of feature selection and different class imbalance strategies on classification results.

In summary, the experiments demonstrate the effectiveness of feature selection, imbalance strategies, and the Bayesian hyperparameter optimization in the classification of network intrusion detection datasets. It has been observed that the proposed intrusion detection systems for wired, wireless, and SDN networks perform well in terms of normal/anomaly detection when hyper parameters are optimized on the validation set and evaluated on the test set. Balanced versions of the datasets yield better performance compared to imbalanced versions. The best performance of network intrusion detection systems (NIDS) has been reported in terms of accuracy, precision, recall, F1 measure, detection rate (DR) and false alarm rate (FAR). The results show that the optimum values for F1 measure, overall accuracy, detection rate, and false alarm rate are high in all three datasets.

5. Conclusion

This study mainly aims to address a significant gap in the literature by evaluating wired, wireless, and software-defined networking (SDN) networks from different perspectives using various state-of-the-art and hybrid machine learning strategies to develop efficient network intrusion detection systems. The focus is on addressing class imbalance problems, performing feature selection and extraction, and conducting binary classification tasks effectively.

Table 6. Binary classification with Bayesian optimization results on the AWID dataset. The bold ones present the best F1-measure and accuracy scores for the AWID dataset and its subsets.

Dataset		Model	Precision	Recall	F1-score	Accuracy
AWID (12 features)	Imbalance	kNN	0.85	0.91	0.879	0.9798
		Random forest	0.97	0.81	0.8828	0.9827
		SVM	0.28	0.99	0.4365	0.7943
		XGBoost	0.94	0.12	0.2128	0.9286
		kNN + RF	0.96	0.32	0.4799	0.9442
		AE + kNN	0.98	0.42	0.588	0.9526
	Balance	kNN	0.84	0.91	0.8736	0.9788
		Random forest	0.96	0.82	0.8845	0.9828
		SVM	0.43	0.43	0.4299	0.9083
		XGBoost	0.95	0.81	0.8744	0.9813
		kNN + RF	0.98	0.84	**0.9046**	**0.9857**
		AE + kNN	0.98	0.42	0.588	0.9526
AWID (20 features)	Imbalance	kNN	0.84	0.86	0.8499	0.9756
		Random forest	0.97	0.66	0.7855	0.971
		SVM	0.63	0.63	0.63	0.9404
		XGBoost	0.97	0.49	0.6511	0.9577
		kNN + RF	0.97	0.65	0.7784	0.9702
		AE + kNN	0.98	0.46	0.6261	0.9558
	Balance	kNN	0.84	0.91	0.8736	0.9788
		Random forest	0.99	0.99	**0.997**	**0.9995**
		SVM	0.43	0.43	0.4299	0.9083
		XGBoost	0.79	0.80	0.795	0.9668
		kNN + RF	0.98	0.82	0.8929	0.9842
		AE + RF	0.98	0.46	0.6261	0.9558

To achieve this, this study utilizes the SMOTE, ADASYN, and Tomek link algorithms for handling class imbalances on wired, wireless, and SDN networking datasets. The eXtreme Gradient Boosting (XGBoost) feature selection methodology has been employed to identify the most informative features. For the binary classification task, several machine learning methods have been applied, including SVM, kNN, XGBoost, random forest, and autoencoder-based ensemble classifiers.

The datasets used in this study are publicly accessible wireless (AWID), wired (UNSW-NB15), and SDN (InSDN) network intrusion datasets. The performance of the proposed intrusion detection systems in this study has been assessed across important conditions, such as with or without feature selection and using imbalanced or balanced datasets.

When optimized on the validation set and evaluated on the test set, this developed models for wired, wireless, and SDN networks perform well in terms of binary classification. Balanced versions of the datasets perform better than their imbalanced counterparts. Table 9 provides a summary of the best performance results for each dataset in terms of F1-measure, overall accuracy, detection rate, and false alarm rate. In addition, Table 1 demonstrates that the combination of the machine learning methodologies proposed in this study generate superior outcomes in comparison to the existing literature.

Table 7. Binary classification with Bayesian optimization results on the UNSW-NB15 dataset. The bold ones present the best F1-measure and accuracy scores for the UNSW-NB15 dataset and its subsets.

Dataset		Model	Precision	Recall	F1-score	Accuracy
UNSW-NB15 (37 features)	Imbalance	kNN	0.84	0.92	0.8782	0.8595
		Random forest	0.82	0.99	0.8955	0.8732
		SVM	0.74	0.97	0.8469	0.803
		XGBoost	0.74	0.86	0.7955	0.7565
		kNN + RF	0.84	0.92	0.8781	0.8595
		AE + RF	0.81	0.99	0.891	0.8666
	Balance	kNN	0.86	0.91	0.8843	0.8689
		Random forest	0.87	0.98	**0.9155**	**0.9017**
		SVM	0.84	0.91	0.8736	0.855
		XGBoost	0.99	0.52	0.6819	0.7328
		kNN + RF	0.94	0.68	0.7891	0.7999
		AE + RF	0.85	0.97	0.906	0.8892
UNSW-NB15 (20 features)	Imbalance	kNN	0.83	0.90	0.8636	0.8434
		Random forest	0.82	0.99	0.8941	0.8711
		SVM	0.74	0.98	0.8469	0.803
		XGBoost	0.76	0.94	0.8405	0.8035
		kNN + RF	0.83	0.90	0.8636	0.8434
		AE + RF	0.81	0.99	0.891	0.8666
	Balance	kNN	0.83	0.91	0.8682	0.8478
		Random forest	0.93	0.94	**0.9357**	**0.9286**
		SVM	0.89	0.85	0.8695	0.8596
		XGBoost	0.98	0.56	0.7127	0.7514
		kNN + RF	0.90	0.71	0.7938	0.7969
		AE + RF	0.86	0.97	0.9117	0.8965
UNSW-NB15 (all features)	Imbalance	kNN	0.84	0.92	0.8781	0.8595
		Random forest	0.82	0.99	0.8923	0.8690
		SVM	0.75	0.99	0.8553	0.8146
		XGBoost	0.78	0.97	0.8647	0.8328
		kNN + RF	0.84	0.92	0.8782	0.8595
		AE + RF	0.81	0.99	0.891	0.8666
	Balance	kNN	0.86	0.91	0.8843	0.8689
		Random forest	0.93	0.94	**0.9356**	**0.9289**
		SVM	0.96	0.84	0.896	0.893
		XGBoost	0.99	0.55	0.7071	0.7492
		kNN + RF	0.96	0.67	0.7892	0.8029
		AE + RF	0.83	0.98	0.8988	0.8785

Importantly, the study demonstrates that reliable NIDS can be generated with oversampling techniques, efficient feature selection techniques, and cost-effective tree-based algorithms. Furthermore, it is noteworthy that the proposed intrusion detection systems provide significant performance, which is almost equivalent to when utilizing 20 features, while only utilizing 12 features on the AWID dataset. (Table 6).

Table 8. Binary classification with Bayesian optimization results on the InSDN dataset. The bold ones present the best F1-measure and accuracy scores for the InSDN dataset and its subsets.

Dataset		Model	Precision	Recall	F1-score	Accuracy
InSDN	Imbalance	kNN	0.99	0.99	**0.9998**	**0.9996**
		Random forest	0.99	0.99	0.9971	0.9953
		SVM	0.99	0.99	0.9982	0.9970
		XGBoost	0.99	0.99	0.9966	0.9946
		kNN + RF	0.99	0.99	0.9998	0.9997
		AE + RF	0.99	0.99	**0.9998**	**0.9996**
	Balance	kNN	0.99	0.99	0.9967	0.9947
		Random forest	0.99	0.99	0.9970	0.9941
		SVM	0.99	0.99	0.9987	0.9979
		XGBoost	0.99	0.86	0.9263	0.8901
		kNN + RF	0.99	0.99	0.9967	0.9947
		AE + RF	0.99	0.99	**0.9998**	**0.9996**

Table 9. The best results obtained from the AWID, UNSW, and InSDN datasets based on different numbers of selected features.

Dataset	No. of selected features	Method	Imbalance strategy	Prec	Rec	FM	Acc	DR	FAR
AWID	12	Voting ensemble (kNN+ RF)	SMOTE + Tomek link	0.98	0.84	0.9046	0.9857	0.9797	0.0151
AWID	20	RF	ADASYN	0.99	0.99	0.997	0.9995	0.9999	0.0171
UNSW-NB15	37	RF	ADASYN	0.87	0.98	0.9155	0.9017	0.8688	0.0464
UNSW-NB15	20	RF	ADASYN	0.93	0.94	0.9357	0.9286	0.9243	0.0678
UNSW-NB15	All	RF	ADASYN	0.93	0.94	0.9356	0.9289	0.9328	0.07597
InSDN	All	AE+ RF	SMOTE + Tomek link	0.99	0.99	0.9998	0.9996	0.9998	0.0012

Acc: accuracy, FM: F measure, Prec: precision, Rec: recall, DR: detection rate, FAR: false alarm rate, RF: random forest.

Overall, this research makes a valuable contribution to the literature in the field of network intrusion detection by offering valuable insights into effective strategies for handling class imbalance, feature selection, and binary classification tasks in wired, wireless, and SDN networks. To the best of our knowledge, no study has been undertaken to evaluate different network intrusion datasets, such as wired, wireless, and SDN, together, considering class imbalance, feature selection, and hyperparameter optimization tasks. The performance results highlight the success of the proposed methods and their potential for practical implementation in real-world network security scenarios. The UNSW-NB15, preprocessed AWID, and InSDN datasets have achieved optimal results based on various metrics including F1-measure, overall accuracy, detection rate, and false alarm rate. The corresponding values for these datasets are [0.9356, 0.9289, 0.9328, 0.07597], [0.997, 0.9995, 0.9999, 0.0171], and [0.9998, 0.9996, 0.9998, 0.0012], respectively. The outcome demonstrates the model's potential capacity for detecting intrusions. However, there are some limitations to the proposed approach. There exists a trade-

off between computational complexity and model performance. Despite achieving high performance models with greater accuracy, it demands substantial resources, especially during the Bayesian optimization step, for detecting intrusions. Consequently, future work should focus on designing effective intrusion detection techniques that classify attacks by accelerating the system using metaheuristics and GPUs.

References

[1] Branco P, Torgo L, Ribeiro R. A survey of predictive modeling on imbalanced domains. ACM Computing Surveys (CSUR) 2016; 49 (2): 1-50. https://doi.org/10.1145/2907070

[2] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 2022; 16: 321-357. https://doi.org/10.1613/jair.953

[3] He H, Bai Y, Garcia EA, Li S. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks. IEEE World Congress on Computational Intelligence; Hong Kong, China; 2008. pp. 1322-1328. https://doi.org/10.1109/IJCNN.2008.4633969

[4] Mustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: IEEE Military Communications and Information Systems Conference (MilCIS); Canberra, ACT, Australia; 2015. pp. 1-6. https://doi.org/10.1109/MilCIS.2015.7348942

[5] Kolias C, Kambourakis G, Stavrou A, Gritzalis S. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials 2015; 18 (1): 184-208. https://doi.org/10.1109/COMST.2015.2402161

[6] Elsayed MS, Le-Khac NA, Jurcut AD. InSDN: a novel SDN intrusion dataset. IEEE Access 2020; 8: 165263-165284. https://doi.org/10.1109/ACCESS.2020.3022633

[7] Potluri S, Diedrich C. Accelerated deep neural networks for enhanced intrusion detection system. In: IEEE 2016 21st International Conference on Emerging Technologies and Factory Automation (ETFA); Berlin, Germany; 2016. pp. 1-8. https://doi.org/10.1109/ETFA.2016.7733515

[8] Hoang TH. Detect Wi-Fi network attacks using parallel genetic programming. In: IEEE 2018 10th International Conference on Knowledge and Systems Engineering (KSE); Ho Chi Minh City, Vietnam; 2018. pp. 370-375. https://doi.org/10.1109/KSE.2018.8573378

[9] Kolukisa B, Dedeturk BK, Hacilar H, Gungor VC. An efficient network intrusion detection approach based on logistic regression model and parallel artificial bee colony algorithm. Computer Standards Interfaces 2023; 89: 103808. https://doi.org/10.1016/j.csi.2023.103808

[10] Kevric J, Jukic S, Subasi A. An effective combining classifier approach using tree algorithms for network intrusion detection. Neural Computing and Applications 2017; 28 (1): 1051-1058. https://doi.org/10.1007/s00521-016-2418-1

[11] Pattawaro A, Polprasert C. Anomaly-based network intrusion detection system through feature selection and hybrid machine learning technique. In: IEEE 2018 16th International Conference on ICT and Knowledge Engineering (ICT&KE); Bangkok, Thailand; 2018. pp. 1-6. https://doi.org/10.1109/ICTKE.2018.8612331

[12] Vaca FD, Niyaz Q. An ensemble learning based Wi-Fi network intrusion detection system (WNIDS). In: IEEE 2018 17th International Symposium on Network Computing and Applications (NCA); Cambridge, MA, USA; 2018. pp. 1-5. https://doi.org/10.1109/NCA.2018.8548315

[13] Dhaliwal SS, Nahid AA, Abbas R. Effective intrusion detection system using XGBoost. Information 2018; 9 (7): 149. https://doi.org/10.3390/info9070149

[14] Logeswari G, Bose S, Anitha T. An intrusion detection system for SDN using machine learning. Intelligent Automation & Soft Computing 2023; 35 (1): 867-880. https://doi.org/10.32604/iasc.2023.026769

[15] Javaid A, Niyaz Q, Sun W, Alam M. A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bioinspired Information and Communications Technologies (formerly BIONETICS); New York, NY, USA; 2016. pp. 21-26. https://doi.org/10.4108/eai.3-12-2015.2262516

[16] Zong B, Song Q, Min MR, Cheng W, Lumezanu C et al. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations; Vancouver, Canada; 2018.

[17] Aygun RC, Yavuz AG. Network anomaly detection with stochastically improved autoencoder based models. In: IEEE 2017 4th International Conference on Cyber Security and Cloud Computing (CSCloud);New York, NY, USA; 2017. pp. 193-198. https://doi.org/10.1109/CSCloud.2017.39

[18] Yousefi-Azar M, Varadharajan V, Hamey L, Tupakula U. Autoencoder-based feature learning for cyber security applications. In: IEEE 2017 International joint conference on neural networks (IJCNN); Anchorage, AK, USA; 2017. pp. 3854-3861. https://doi.org/10.1109/IJCNN.2017.7966342

[19] Zhang B, Yu Y, Li J. Network intrusion detection based on stacked sparse autoencoder and binary tree ensemble method. In: IEEE 2018 International Conference on Communications Workshops (ICC Workshops); Kansas City, MO, USA; 2018. pp. 1-6. https://doi.org/10.1109/ICCW.2018.8403759

[20] Javaid A, Niyaz Q, Sun W, Alam M. A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bioinspired Information and Communications Technologies (formerly BIONETICS); New York, NY, USA; 2016. pp. 21-26. https://doi.org/10.4108/eai.3-12-2015.2262516

[21] Dahiya P, Srivastava DK. Network intrusion detection in big dataset using Spark. Procedia Computer Science 2018; 132: 253-262. https://doi.org/10.1016/j.procs.2018.05.169

[22] Wheelus C, Bou-Harb E, Zhu X. Tackling class imbalance in cyber security datasets. In: IEEE 2018 International Conference on Information Reuse and Integration (IRI); Salt Lake City, UT, USA; 2018. pp. 229-232. https://doi.org/10.1109/IRI.2018.00041

[23] Abdulhammed R, Faezipour M, Abuzneid A, AbuMallouh A. Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. IEEE Sensors Letters 2018; 3 (1): 1-4. https://doi.org/10.1109/LSENS.2018.2879990

[24] Ran J, Ji Y, Tang B. A semi-supervised learning approach to IEEE 802.11 network anomaly detection. In: 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring); Kuala Lumpur, Malaysia; 2019. pp. 1-5. https://doi.org/10.1109/VTCSpring.2019.8746576

[25] Abdelkhalek A, Mashaly M. Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning. The Journal of Supercomputing 2023; 79 (10): 10611-10644. https://doi.org/10.1007/s11227-023-05073-x

[26] Qazi EUH, Imran M, Haider N, Shoaib M, Razzak I. An intelligent and efficient network intrusion detection system using deep learning. Computers and Electrical Engineering 2022; 99: 107764. https://doi.org/10.1016/j.compeleceng.2022.107764

[27] Gupta SK, Tripathi M, Grover J. Hybrid optimization and deep learning based intrusion detection system. Computers and Electrical Engineering 2022; 100: 107876. https://doi.org/10.1016/j.compeleceng.2022.107876

[28] Qazi EUH, Faheem MH, Zia T. HDLNIDS: hybrid deep-learning-based network intrusion detection system. Applied Sciences 2023; 13 (8): 4921. https://doi.org/10.3390/app13084921

[29] Bhayo J, Shah SA, Hameed S, Ahmed A, Nasir J et al. Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks. Engineering Applications of Artificial Intelligence 2023; 123: 106432. https://doi.org/10.1016/j.engappai.2023.106432

[30] Tayfour OE, Marsono MN. Collaborative detection and mitigation of DDoS in software-defined networks. The Journal of Supercomputing 2021; 77: 13166-13190. https://doi.org/10.1007/s11227-021-03782-9

[31] Yoo S, Kim S, Kim S, Kang BB. AI-HydRa: advanced hybrid approach using random forest and deep learning for malware classification. Information Sciences 2021; 546: 420-435. https://doi.org/10.1016/j.ins.2020.08.082

[32] Elsayed MS, Le-Khac NA, Dev S, Jurcut AD. Network anomaly detection using LSTM based autoencoder. In: Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks; Alicante, Spain; 2020. pp. 37-45. https://doi.org/10.1145/3416013.3426457

[33] Firdaus D, Munadi R, Purwanto Y. DDoS attack detection in software defined network using ensemble K-means++ and random forest. In: 2020 IEEE 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI); Yogyakarta, Indonesia; 2020. pp. 164-169. https://doi.org/10.1109/ISRITI51436.2020.9315521

[34] Elsayed MS, Jahromi HZ, Nazir MM, Jurcut AD. The role of CNN for intrusion detection systems: an improved CNN learning approach for SDNs. In: Perakovic D, Knapcikova L (editors). Future Access Enablers for Ubiquitous and Intelligent Infrastructures. FABULOUS 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 382. Cham, Switzerland: Springer, 2021, pp. 91-104. https://doi.org/10.1007/978-3-030-78459-1_7

[35] Tian Q, Han D, Li KC, Liu X, Duan L et al. An intrusion detection approach based on improved deep belief network. Applied Intellgidence 2020; 50: 3162-3178. https://doi.org/10.1007/s10489-020-01694-4

[36] Jing D, Chen HB. SVM based network intrusion detection for the UNSW-NB15 dataset. In: 2019 IEEE 13th International Conference on ASIC (ASICON); Chongqing, China; 2019. pp. 1-4. https://doi.org/10.1109/ASICON47005.2019.8983598

[37] Kumar R, Malik A, Ranga V. An intellectual intrusion detection system using Hybrid Hunger Games Search and Remora Optimization Algorithm for IoT wireless networks. Knowledge-Based Systems 2022; 256: 109762. https://doi.org/10.1016/j.knosys.2022.109762