International Journal of Advanced Technology and Engineering Exploration, Vol 11(112)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2023.10101620

Research Article

Enhancing intrusion detection with imbalanced data classification and feature selection in machine learning algorithms

S.V. Sugin and M. Kanchana*
Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Chennai-603203, Tamilnadu, India

Received: 21-April-2023; Revised: 23-February-2024; Accepted: 26-February-2024

Abstract
The effectiveness of an organization in detecting and preventing computer network (CN) attacks is significantly influenced by the performance of intrusion detection systems (IDS) and intrusion prevention systems (IPS). This research focuses on IDS based on machine learning (ML), asserting that ML-based IDS are effective and accurate in detecting network attacks. The study examines the UNSW-NB15 network IDS dataset, which is used for training and testing the models. Furthermore, a filter-based attribute reduction approach was implemented using the extreme gradient boosting (XGBoost) algorithm. The condensed feature space then facilitates the application of various methods including support vector machine (SVM), logistic regression (LR), k-nearest neighbour (KNN), decision tree (DT), and convolutional neural network (CNN). A suitable feature selection approach is essential to eliminate features with minimal impact on the classification process. Additionally, the study notes that many ML-based IDS suffer from limited identification accuracy and a higher false positive rate (FPR) when trained on highly imbalanced datasets. The research considers configurations for both binary and multiclass classification. Results indicate that the XGBoost based attribute selection approach allows techniques such as DT to enhance the test accuracy of the binary-classification scheme from 88.13% to 90.85%. Moreover, the XGBoost-KNN and XGBoost-DT configurations demonstrate improved performance.

Keywords
Machine learning (ML), Intrusion detection system (IDS), UNSW-NB15 dataset, XGBoost algorithm, Convolutional neural network (CNN).

1.Introduction
Hackers have evolved at a faster rate in terms of their capabilities because of precise technological developments like networks and communication systems. These crooks are constantly looking for new ways to jeopardize the security of computer networks (CN). Intrusion detection systems (IDS) consequently automatically become important parts of a CN. An IDS is a piece of software or hardware that tracks a company's network for potential threats. Meanwhile, an IDS is capable of reacting to and reporting any fraudulent activity. Node-based or host-based IDS (HIDS) [1, 2] Network based IDS (NIDS) or distributed-based IDS (DIDS), and hybrid-based IDS (HYIDS) are the three basic categories into which IDSs fall based on the unique IDS operation concept, this classification was created.

The primary fundamental design goals of IDS are a decrease in false positive (FP) alerts and an increase in detection accuracy. Subsequently, when designing and implementing any IDS, that perspective must be taken into consideration [3]. In currently decades, machine learning (ML)-based IDS have taken over as the industry standard. According to ML systems may now potentially learn from the past and improve. Broadly speaking, the two ML philosophies of supervised ML and unsupervised ML can be separated. Models are learned using labeled data in supervised ML [4]. Unsupervised ML uses unstructured data to train models.

The multiclass and binary classification objectives are carefully taken into consideration when using supervised ML techniques in this study. The classification process occurs whenever a supervised ML technique is asked to identify a specific quality. The training data for the techniques in this configuration frequently has large sizes and high-

*Author for correspondence

dimensional value spaces. Supervised ML techniques require a while time to test and train, according to the quantity of the data. To reduce the quality of attributes needed for the testing and training stages, feature engineering operations must be carried out. A feature reduction method inspired by filters is applied to the work presented in this research, taking into account the features produced and important metrics produced by the XGBoost algorithm [5].

K-nearest neighbour (KNN), logistic regression (LR), support vector machine (SVM), convolutional neural network (CNN), and decision tree (DT) are examples of supervised ML techniques for IDS that we used in our experimental methods. The dataset UNSW-NB15 is used [6, 7] and the XGBoost approach to build a reduced and ideal feature vector by computing the feature importance measure for each feature. Additionally, for each of the chosen ML algorithms, both the binary and multiclass classification systems have been included in our performance study [8]. Supervised ML techniques, especially those applied to binary and multi-class classification tasks are examined in this research. When a supervised ML model is assigned the duty of predicting a discrete value, the classification operation takes place [9]. Large datasets with a high-dimensional feature space are typically used to train the models in this setup. Training and testing supervised ML models can be time-consuming because of their complexity. In order to reduce the number of attributes needed for the testing and training stages, feature engineering procedures are crucial. The work discussed in this paper considers a filter inspired attributes reduction method that makes use of the feature relevance measurements produced by the XGBoost method. When you use the filter-inspired feature extraction approach instead of the wrapper based feature extraction technique, the feature space reduction step is done separately from the classifier that makes the final predictions [10]. In addition, a comprehensive literature analysis is carried out, and the outcomes of this study are contrasted with those of the numerous ML approaches that were surveyed.

The key objective of this work is to classify the intrusion by selecting features based on their classification accuracy. The motivation of the work is to perform binary and multi-class classifications on the different ML methods. The contributions of this original study summary use the following: First, a description of appropriate work is given. A summary of all the ML methods applied in this study is given. Furthermore, information is provided regarding the UNSW-NB15 dataset. Similarly, all of the algorithms used during the experimental methods are detailed, along with the recommended IDS strategy. The experimental procedures and discussions are then all compiled into a single presentation.

The organization of the paper into the following sections: The literature review is described in section 2. The methodology presented in section 3. The section 4 discusses the results, and section 5 shows the findings of the results analysis. Section 6 covers the conclusion and further work.

2.Literature review

The KDDCup99 and UNSW-NB15 network datasets [11] were used in the implementation of the genetic algorithm (GA) and LR, wrapper-based attribute choosing approaches by the researchers [12]. The findings indicate that after numerous simulations were performed, the DT classifier with GA and LR in combination was able to achieve an identification outcome of 81.42% and 6.39% for the false alert rate (FAR) by using 20 of the 42 attributes available in the UNSW-NB15 dataset attribute vector. Using 18 features, the GA-LR and DT classifiers were able to recognize the KDDCup99 dataset with an identification value of 99.90% and a FAR value of 0.11%.

The filter-based solution for distributed denial of service (DDoS) identification relief factor reported by the authors in [13] utilized a number of filters: information gain (IG), gain ratio, chi-square, and relief are three filtering methods. The researchers used the dataset for network security dataset (NSL-KDD) attack identification to demonstrate how well the system worked. The DT algorithm was used by the authors for classification, and k = 10 cross validation was used to train and validate the algorithm. The trial results showed that the DT-classifier only used 13 out of the 42 attributes to achieve an identification accuracy rate of 99.67% and a FAR of 0.42%. The study, however, did not go into considerable length on the multi-class prediction issue of the NSL-KDD dataset.

In [14], IDS is described in conjunction with a filter inspired input reduction technique. For assessment in this study, multiple datasets, like the KDDCup99, the NSL-KDD, and the Kyoto2006, were used. The authors of this research looked into the flexible mutual information (FMI) technique to make claims about potential correlations between various input variables. A non-linear correlation metric is the FMI.

The least square SVM (LS-SVM) classifier was employed in the experiments. The outcomes demonstrated that the LS-SVM-FMI for the NSL-KDD with 18 characteristics had a FAR of 0.28% and an accuracy of 99.94%. In the case of KDDCup99, SVM, FMI, and LS achieved an average accuracy of 79.16%, while in the instance of Kyoto2006, SVM, FMI, and LS achieved a 97.50% identification value and a 0.43% FAR rate on iteration 10.

To reduce the quantity of input qualities essential for the testing and training of their model, the authors of [15] built IDS. The correlation input selection method was combined with the DT classifier. The experimental approaches make use of the NSL-KDD dataset. The set of features was given the filter, and 14 features were chosen [16, 17]. The author also considered both the multi and binary classification setup options, which included each of the five NSL-KDD attack types. The outcomes of the trial showed that the system had a multiclass configuration accuracy of 83.66% and a binary configuration accuracy of 90.30%.

Albasheer et al. [18] proposed IDS that makes use of a two stage (TS) classifier approach that is according to the random forest (RF) classifier. The criteria for the binary classification process's required attributes were chosen using the IG approach. The UNSW-NB15 dataset's categorical features were changed by the authors using the one-hot encoding technique. The first part of the IG work TS involves the identification of minority classes, while the second stage involves the identification of majority classes. The ultimate prediction is then made using the combined data from each stage [19]. The accuracy and the FAR were employed by the researchers as outcome indicators. After the multiple tests, the TS and IG obtained a FAR of 15.78% and an accuracy of 85.78%.

In [20], the researchers used the pigeon inspired optimizer (PIO) to build a feature reduction strategy for IDS. The PIO algorithm draws inspiration from biological processes, namely the interactions between white pigeons and their prey. These birds are continually adjusting by comparing their fighting stance to the finest bird in the flock [21]. The two PIO kinds that were examined in this paper were sigmoid and cosine PIO. Three intrusion detection sets were taken into consideration: UNSW-NB15, NSL-KDD, and KDDCup99. The sigmoid-PIO suggested 10 features from the KDDCup99, and the cosine-PIO selected seven of the NSL-KDD features. The sigmoid-PIO chooses 18 characteristics and the cosine-PIO chooses 5 of the UNSW-NB15 features, the sigmoid-PIO chooses 14 characteristics, and the cosine-PIO chooses 5. The Sigmoid PIO's accuracy was 90.3% on the UNSW-NB15, 87.21% over the NSL-KDD, and 94.7% utilizing the KDDCup99. Conversely, cosine PIO scored 92.07% on UNSW-NB15, 89.32% on NSL-KDD, and 97.20% on KDDCup99.

Ortega-fernandez et al. [22] constructed a variety of attribute selection techniques using the UNSW-NB15 in an approach to select the best feature space. The following techniques were put into practice with the help of the weka tool: the ranker method, the greedy stepwise, the attribute evaluator, and IG. Two subsets were taken into consideration after different simulations. The authors used the kappa statistic metric to determine each subset's effectiveness. In spite of the fact that a number of classifiers were taken into consideration during the trials, the RF-classifier was chosen as the overall best approach. The first subset, which contained eight significant features over the test dataset, obtained a kappa score of 0.789 and an accuracy of 76.672%. The accuracy was 82.627%, and the kappa value was 0.774 for the second group, which only included five significant features.

An IDS system constructed by Sah et al. [23] was verified with the UNSW-NB15 dataset. The authors of this study examined a feature reduction strategy that drew inspiration from the IG methodology, and 22 key qualities were chosen as a consequence of this filter-based feature extraction technique. Furthermore, to perform the classification process, the IDS presented in this research employed a combined rule based model that made use of several tree-based classifiers. The attack accuracy on the test data, the F-Measure (FM), and the FAR were the primary metrics used to assess the system's performance. The suggested IDS received an accuracy of 56.02%, a FM of 91%, and a FAR of 2.13%, according to the outcomes. When different ML algorithms are taken into account and substituted for strict compliance with tree-based methodologies, the research findings can be further enhanced.

In [24] showed how to use IDS based CNN to determine imbalanced congestion in the network. A performance rate was predicted using indicators like detection rate (DR), accuracy and FAR. The accuracy of the IDS-based CNN model was approximately

84.21%. This approach increased the DR of user to root (U2R) and remote to local (R2L) attacks. The system performed poorly as a result of the minimum classification accuracy that was reached.

Sekhar et al. [25] used a fruit fly optimization-based deep auto encoder to explain an intrusion detection process. One of this paper's primary objectives was to secure the network and stop hacker attacks. Using the UNSW-NB15 and NSL-KDD datasets, the experiment was assessed and verified. Comparing this method to other ML techniques, the accuracy was greater.

In order to reduce the quantity of noisy data records within the majority classes, Ahmadi et al. [26] proposed an NIDS structure that made use of the one side selection (OSS) technique. Furthermore, the researchers employed a synthetic methodology known as synthetic minority over sampling (SMOS) to augment the quantity of minority samples inside the dataset. Additionally, CNN was used to extract the spatial attributes, while bi-directional long-short-term memory (bi-LSTM) models were used to select the temporal attributes. The researchers thought that the accuracy of the test data was the most important performance factor. To test the suggested framework, they used the NSL-KDD and UNSW-NB15 intrusion detection datasets. The outcomes suggested that the CNN bi-LSTM obtained accuracy for each dataset of 78.17% and 84.59%, respectively.

Even though a wide range of specialists put forth a large number of hybrid models, they mostly focused on recognized attacks and attempted to merge any two models. Furthermore, the suggested model is evaluated and trained on a single dataset, showing superior accuracy on that specific dataset alone. Few specialists have looked into hybrid approaches based on techniques for detecting intrusions. Although hybrid approaches were employed in several previous works, the binary classification accuracy was higher. Another significant flaw in the current models is that, rather than using UNSW-NB15 datasets explicitly, the majority of experts used standard network intrusion datasets. The suggested model, which focuses on a hybrid intrusion detection technique for network security, was developed based on the conclusions drawn from the literature study in order to address the shortcomings of the current model. For both anomaly and signature-based attacks, the suggested hybrid model combines the XGBoost algorithm with a hybrid detection technique. Using the UNSW-NB15 dataset, the model was evaluated. The suggested model outperformed other models in accuracy even after arbitrarily reducing the total number of features in each dataset.

3.ML methods

The following algorithms are used for model training and validation.

3.1SVM

SVM are the most adaptable ML models and can do both regression and classification tasks. One of the prevalent techniques in ML research is SVM. To determine the optimum hyper planes, the SVM technique attempts to classify a dataset, which is divided into various groups. One benefit of utilizing SVM is that it often performs perfectly for elevated dimensional input spaces [27].

3.2LR

Although LR is a ML method that is mainly used in binary classification tasks. When the learning method uses one-vs-rest approaches for LR can also be used for multiclass classification tasks. A linear ML model is subjected to the sigmoid function or one of its modifications in the LR model. This method produces a result that is reduced between (0, 1). The probability of a specific class is determined by an output that is closer to 1. Equations 1, 2, and 3 in the mathematical formulation [28, 29] provide a description.

Linear Model:

$$y = b + w_1 x_1 + w_2 x_2 + \ldots\ldots + w_n x_n \quad (1)$$

SE: $\sigma(k) = \frac{1}{1 + e^{-k}}$ (2)

LR:$\sigma(y) = \frac{1}{1 + e^{-y}}$ (3)

3.3CNN

An approach to ML known as a CNN simulates how biological neurons function internally. Node, or neuron, is the name for the most fundamental element of a CNN. It is suggested that a perceptron is a CNN with a single hidden input layer of processing to produce an output. Multilayer perceptrons (MLP) or feed forward CNN are terms that have been used to describe CNNs that have many hidden layers [30]. An activation function such as the sigmoid expression (SE), $\sigma(k) = \frac{1}{1 + e^{-k}}$ a rectified linear unit (ReLU), F(k) = Max(0, k), a hyperbolic tangent $tanh(t) = \frac{1 - e^{-2t}}{1 + e^{-2t}}$ or a combination of those functions, *Figure 1* is used to process neurons in a CNN.

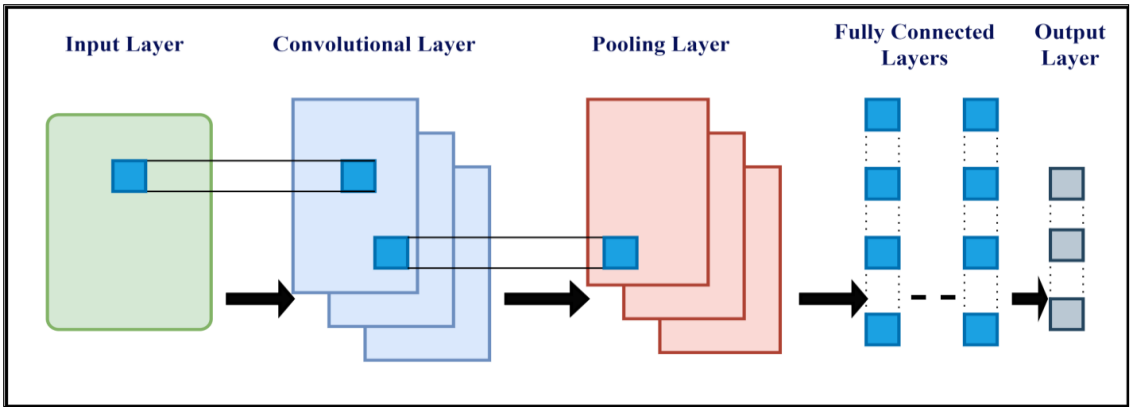

Figure 1 CNN model

3.4DT

In order to do classification and regression tasks, DT is used [31], which is a supervised ML technique. The decision making process leads to the creation of a tree like model. DT is now simpler for the user to understand. The display of the produced trees is also possible thanks to a variety of ML tools.

3.5KNN

Both supervised and unsupervised procedures can be carried out using the KNN, ML technique [32]. For instance, there are several clustering algorithms being used currently that are based on KNN. In this study, we applied the supervised ML variant of the KNN algorithm. This method is based on the common euclidean metric (EM). Two point's instances in a space are separated by the EM. The EM within (m, n) and Δ (m, n) is calculated as shown in Equation 4:

$$\Delta(m,n) = \sqrt{\sum_{i=1}^{r} (y_n - y_n)^2} \qquad (4)$$

3.6UNSW-NB15 network dataset

A total of above 2.5 million network packets were used to simulate the dataset when it was made available in 2015 [33]. The original traffic in the UNSW-NB15 dataset was developed by Cyber Range Lab at UNSW Canberra. They used the IXIA perfect storm program to combine real-world events with fabricated attacks. There are a total of 2540044 records in four various comma separated value (CSV) files. Along with nine other attack types, this data set also contains non-anomalous packets (exploits, denial of service, reconnaissance, generic, shellcode, fuzzers, backdoors, worms, and analysis). As a result of 87.01% of the packets being non anomalous, the dataset is seriously skewed. Additional details about this dataset may be found in *Table 1* of the packet distribution. The UNSW-NB15 dataset is split into the following two primary sets: UNSW-NB15: Test (100%), used to test the trained models, and UNSW-NB15: Train, used to train a variety of models. In our research, the UNSW-NB15: Train was further split into two parts: UNSW-NB15: Train (75% of the whole training set) is used for training, while UNSW-NB15: Validation (25% of the whole training set) is used for validation before testing. When employing that technique, it is essential to prevent a model from being trained on the test set or evaluation, as this could result in data losses. Data losses happen as part of the training phase when a model gets data that it shouldn't and introduces bias into the final model [34, 35]. As a result, the model consequently underperforms on earlier untested data.

The following categories of network attacks are represented in the UNSW-NB15 network dataset instances: backdoors, reconnaissance, fuzzers, shellcode, analysis, generic, DoS, worms, shellcode, and exploits. *Table 2* also gives relevant data on the characteristics of each attack type and the distribution of values across the data subsets [36].

The UNSW-NB15 dataset is what we use for our research methodology [37]. The UNSW-NB15 dataset has 42 attributes in its clean format, as indicated in *Table 3*. The approach to choosing features based on XGBoost is shown in *Table 4*. The result was a decreased feature-vector with 19 attributes. *Table 4* shows that based on the selected features for the dataset UNSW-NB15, we calculated the importance score of the feature-vector.

In *Figure 2*, more information is provided on the system architectural design that is suggested in this study. Data processing takes up the entire first block; this method is frequently referred to as data engineering [38].

Table 1 Dataset record distribution

Types	No. Records	% of total data	Description
Normal	2218761	87.35	Natural transactional data
Fuzzers	24246	0.95	Supplying a programme or network with randomly generated data in an attempt to interrupt operation.
Analysis	2677	0.11	It includes various message, penetration of HTML tags, and attack by specific port.
Backdoors	2329	0.1	A method by which a protection system defence is secretly disregarded to get access to a system or its information.
DoS	16353	0.64	A malware attack to prevent people by making use of a network resource or server, typically by briefly stopping services or suspending offered by an internet connected host.
Exploits	44525	1.75	There is a security issue in an operating system or software, which the attacker is aware of and uses by exploiting the weakness.
Generic	215481	8.48	A method is applicable to all the block crypto algorithms (with a specific key and byte sizes) without taking the cipher's model into account.
Reconnaissance	13987	0.55	Includes all Strikes that can be used to predict information gathering attacks.
Shellcode	1511	0.06	A brief segment of code that is used to exploit a software defect.
Worms	174	0.01	To infect further methods, the malware replicates itself. It usually disseminates itself over a CN and gains access to the target computer by exploiting security flaws.

Table 2 Restructured UNSW-NB15 network dataset instances

Attack	Data	Training Data	Validation Data	Test Data
Normal	56000	41911	14089	37000
Fuzzers	18184	13608	4576	6062
Analysis	2000	1477	523	677
Backdoors	1746	1330	416	583
DoS	12264	9237	3027	4089
Exploits	33393	25034	8359	11132
Generic	40000	30081	9919	18871
Reconnaissance	10491	7875	2616	3496
Worms	130	99	31	44
Shellcode	1133	854	279	378

Table 3 UNSW-NB15 features list

Feature_No	Feature	Type	Feature_No	Feature	Type
FN_1	dur	float	FN_22	dtcpb	integer
FN_2	proto	categorical	FN_23	dwin	integer
FN_3	service	categorical	FN_24	tcprtt	float
FN_4	state	categorical	FN_25	synack	float
FN_5	spkts	integer	FN_26	ackdat	float
FN_6	dpkts	integer	FN_27	smean	integer
FN_7	sbytes	integer	FN_28	dmean	integer
FN_8	dbytes	integer	FN_29	trans_depth	integer
FN_9	rate	float	FN_30	response_body_len	integer
FN_10	sttl	integer	FN_31	ct_srv_src	integer
FN_11	dttl	integer	FN_32	ct_state_ttl	integer
FN_12	sload	float	FN_33	ct_dst_ltm	integer
FN_13	dload	float	FN_34	ct_src_dport_ltm	integer
FN_14	sloss	integer	FN_35	ct_dst_sport_ltm	integer
FN_15	dloss	integer	FN_36	ct_dst_src_ltm	integer
FN_16	sinpkt	float	FN_37	is_ftp_login	integer
FN_17	dinpkt	float	FN_38	ct_ftp_cmd	integer
FN_18	sjit	float	FN_39	ct_flw_http_mthd	integer
FN_19	djit	float	FN_40	ct_src_ltm	integer

Feature_No	Feature	Type	Feature_No	Feature	Type
FN_20	swin	integer	FN_41	ct_src_dtm	integer
FN_21	stcpb	integer	FN_42	is_sm_ips_ports	binary

Table 4 Selected features for the UNSW-NB15 network dataset

Feature_No	Feature	Type	Feature-importance-Score
FN10	sttl	integer	0.80337
FN41	ct_srv_dst	integer	0.03939
FN7	sbytes	integer	0.03738
FN27	smean	integer	0.01988
FN2	proto	integer	0.01885
FN32	ct_state_ttl	integer	0.01678
FN14	sloss	integer	0.01201
FN25	synack	float	0.01013
FN35	ct_dst_src_ltm	integer	0.00720
FN28	dmean	integer	0.00713
FN31	ct_srv_src	integer	0.00675
FN3	service	categorical	0.00631
FN13	ct_dst_sport_ltm	integer	0.00372
FN8	dbytes	integer	0.00271
FN15	dloss	integer	0.00178
FN4	state	categorical	0.00155
FN24	tcprtt	float	0.00123
FN34	ct_src_dport_ltm	integer	0.00053
FN9	rate	float	0.00051

A successful learning process depends on this step. Cleaning, feature selection, and normalization are the three stages of data- processing. A filter based approach to feature selection is used, which draws inspiration from the XGBoost algorithm for calculating feature importance scores. The following step involves training the model using the training set after the required feature vector has been selected. After training, a model is identified using the set of validations. Therefore, the validated model is applied to the test using the test dataset [39]. A fine-tuned and suitable model is discovered after repeating the above mentioned process.

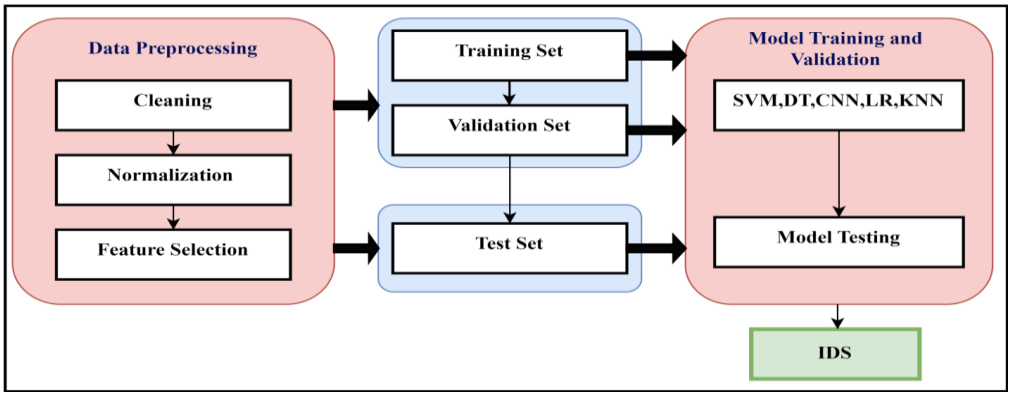

Figure 2 ML based IDS architecture

3.7Data preprocessing

Since it includes both data preparation and data transformation from raw data, this is one of the most crucial steps in the data mining process. The actual data must be translated into an acceptable format that is appropriate for researching prior to intrusion categorization. To improve the classifier's efficiency, various pre-processing approaches such as cleaning, normalization, and feature selection are used [40].

3.8Feature selection

Selecting acceptable characteristics after data preprocessing is regarded as an important stage in an IDS. The attribute selection procedure eliminates

unnecessary features, splits redundant features, and separates noisy data, providing only the necessary features.

3.9Feature normalization

The improved values for numerals of several attributes have an impact on the development of a suitable and fine-tuned model using ML approaches like SVM, LR, CNN, and KNN [41, 42]. It also takes a lot of processing resources to train high dimensional datasets. Data is regularly scaled using strategies like standardization of max normalization, Z-score, decimal scaling, and easy-difficult set ability to address these problems. In many cases, the application indicates which technique to take. Our processing data phase employs easy-difficult set scaling (Equation 5).

Easy-Difficult scaling of attributes f:

$$f_{norm} = \frac{f - f_{easy}}{f_{difficult} - f_{easy}} \quad (5)$$

Table 5 (Algorithm 1) describes how the standardization calculation is carried out provide a dataset with an input vector (feature space) defined by U (f1, ... , fn), 1 < n < N, where N is the whole number of occurrences (features) in the space. The XGBoost algorithm's core idea is to train an ensemble tree model incrementally while also including a penalty parameter to lessen the model's complexity [43, 44]. It is best to use a small expression in Equation 6. An explanation of the mathematical formulation is provided below:

The objective function is therefore written as follows: Let $\hat{a}^{(t)}$ be the prediction of the k^{th} case at the t^{th} increment.

$$L^{(t)} = a_0 + \sum_{k=1}^{n} l\left(a_k, \hat{a}_k^{k-1} + f_t(X_i)\right) + \theta(f_t) \quad (6)$$

L represents the loss function. The XGBoost technique objective is to minimize the second-order expression below using the simplified form:

$$L'^{(t)} = a_0 + \sum_{k=1}^{n} \left(\left[g_k f_t(X_k) + \frac{1}{2} h_k f_t^2(X_k) \right] + \theta(f_t) \right) \quad (7)$$

where the letters g_k and h_k, respectively, stand for the gradients first and second order expressions in relation to the loss-function. Given a dataset, the XGBoost technique is capable of calculating scores for each feature. The term "feature importance" refers to this score metric. In this study, the importance of each input for the learning and classification processes is determined using the FI measure [45, 46]. Algorithm 2 (*Table 6*) provides a description of the algorithm illuminating the XGBoost technique for producing the attribute importance by means of scikit-learn.

3.10Performance metrics

ML-based IDS systems can be evaluated using a variety of metrics, but the goal of this research is to increase the number of cases in the test dataset that are properly predicted. The important indicator to think about is the accuracy described here in Equation 8:

$$Accuracy = \frac{TN + TP}{FP + FN + TP + TN} \quad (8)$$

where true positive (TP) denotes the proportion of incidents that were properly described as attacks. True negative (TN) is the proportion of genuine data that is identified as valid. The (%) percentage of legitimate traffic that is classified incorrectly as an attack is known as a FP, also known as a Type-I error. False negative (FN), often indicated to as Type-II error, is the proportion of genuine data that is labeled as incursions. Other metrics that we evaluate in this study include recall, precision, and F1-Score. These terms are described below in Equation 9.

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1score = 2 \times \frac{Precision.Recall}{Precision + Recall} \quad (11)$$

The harmonic ratio of the expression in (Equation 9) is what is used to calculate the F1-Score (Equation 10). Likewise, certain applications are made to focus on precision or recall. The F_β specified below can be modified to do that during training:

$$F_\beta = \frac{(1 + \beta)^2 \times TP}{(1 + \beta)^2 TP + \beta FN + FP} \quad (12)$$

where the element β can be changed depending on the evaluation. But in this study, we employed the F1-Score, a compromise between recall and precision.

A detailed list of abbreviations is included in *Appendix I.*

Table 5 Easy-difficult set scaling algorithm

Algorithm 1:Easy-Difficult set scaling algorithms

Input: U (f_1,….,f_n) where 1 < n < N
Output: $U_{normalized}(\int_n^{norm}, \dots\dots, \int_n^{norm}.)$
for(j from 1 to k)**do**
if(f_ja easy set value) **then**
Step 1: Encode using one hot encode attribute conversion.
Step 2:Compute Easy-Difficult set Scaling $\int_n^{norm} = \frac{f_{n-(f_n)easy}}{(f_n)difficult-(f_n)easy}$
end if
Step 1:Compute Easy-Difficult set Scaling $\int_n^{norm} = \frac{f_{n-(f_n)easy}}{(f_n)difficult-(f_n)easy}$
end for

Table 6 XGBoost Algorithm

Algorithm 2:XGBoost Algorithm

Input : $U_{normalized}(\int_1^{norm}, \dots\dots, \int_n^{norm}.)$
Output:$U_{optimal}$: the identified attributes vector
Step 1 : Resolve the normalised attributes vector.
Step 2 : Consider a custom dictionary ***S*** to save the outcomes.
Step 3 : Initiate a Gradient Boosting Classifier C_f
Step 4 : fit C_f
Step 5 : Generate FI_s
Step 6 : Determine the **FI** threshold FI_p
Step 7 :
for (n from $U_{normalized}$**) do**
if(FI (x^i) $\geq$ FI_p) then
append FI (x^i) into *S*
end if
end for
Step 8 : Utilize S value to achieve $U_{optimal}$

4.Results

CNN, KNN, SVM, LR, and DT were among the ML techniques that were taken into consideration for our research strategy, which was split into two primary phases. In the initial phase of the tests, both the binary values and the multiclass setup made use of the entire UNSW-NB15 feature space (42 features). The XGBoost-based feature selection technique was introduced in the next stage. Following that, a reduced vector of features with 19 properties was created (given in *Table 4*). Researchers ran the tests that included both the multiclass and the binary-classification techniques, which use the best feature vector. *Tables 7*, *8*, *9*, and *10* show the results of our experiments. *Table 7* shows the results of the binary classifying method using the whole attributes section of the UNSW-NB15 dataset and the ML methods. *Table 9* displays the outcomes of ML techniques for the restricted vector of features binary-classification method. The outcomes of the ML techniques for the multiclass classifying problem utilizing the complete attribute space and the decreased attribute vector are appropriately presented in *Tables 8* and *10*. The accuracy found using training data is indicated by the training accuracy in each table. Testing accuracy signifies the accuracy acquired on test data, while validation accuracy indicates the accuracy detected in the validation data.

Table 7 Result with binary-classification using 42-attributes

ML Algorithm	Training data accuracy (%)	Validation data accuracy (%)	Test data accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	94.49%	94.21%	86.71%	81.54%	98.06%	89.04%
LR	93.22%	92.87%	79.59%	73.32%	98.94%	84.22%
SVM	70.98%	70.63%	62.42%	60.91%	88.58%	71.18%
KNN	96.76%	93.60%	83.19%	79.15%	94.30%	86.06%
DT	93.65%	93.37%	88.13%	83.91%	96.47%	90%

Table 8 Result with multiclass-classification using 42-attributes

ML Algorithm	Training data accuracy (%)	Validation data accuracy (%)	Test data accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	79.91%	79.61%	75.62%	79.92%	75.61%	76.58%
LR	75.51%	73.93%	65.53%	76.91%	65.54%	66.62%
SVM	53.43%	52.67%	61.09%	47.47%	62.00%	53.77%
KNN	81.75%	76.83%	70.09%	75.79%	70.21%	72.03%
DT	77.69%	77.38%	66.03%	79.82%	66.04%	51.12%

Table 9 Result with binary-classification using 19-attributes

ML Algorithm	Training data accuracy (%)	Validation data accuracy (%)	Test data accuracy (%)	Precision-(%)	Recall-(%)	F1-Score (%)
CNN	93.75%	93.66%	84.39%	78.56%	98.53%	87.42%
LR	89.21%	89.25%	77.64%	73.18%	93.74%	82.20%
SVM	75.42%	75.51%	60.89%	58.89%	95.88%	72.97%
KNN	95.86%	94.73%	84.46%	80.31%	95.09%	87.08%
DT	94.12%	93.81%	90.85%	80.33%	98.38%	88.45%

Table 10 Result with multiclass-classification using 19-attributes

ML Algorithm	Training data accuracy (%)	Validation data accuracy (%)	Test data accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	79.46%	78.91%	77.51%	79.50%	77.53%	77.28%
LR	72.53%	71.81%	65.29%	70.88%	65.29%	65.96%
SVM	53.60%	52.97%	61.53%	53.95%	61.52%	51.31%
KNN	82.66%	79.87%	72.30%	77.24%	72.30%	73.81%
DT	78.75%	78.43%	67.57%	79.66%	67.56%	69.26%

The outcomes of our experimental procedures are shown in *Tables 7*, *8*, *9*, and *10*. The outcomes of the training data accuracy, compared to other ML models, show that the KNN algorithm training level is increasing. The CNN algorithm increases the accuracy of the testing data. *Tables 7*, *8*, *9*, and *10* show the results of binary and multiclass classification using 19, 42 attributes.

Regarding CNNs, all of our studies utilized the Adam solver rather than the conventional stochastic gradient descent (SGD) method because it is a SGD-based approach that is ideal for huge sets of data. Only one hidden layer that might contain the applied amount of neurons was used to create the CNNs; the concealed layer size ranges from 5 to 100. The rate of learning was adjustable. As shown in *Table 7*, the CNN network that performed best used 150 neurons and a flexible learning speed of 0.02 to attain an accuracy rating throughout the test set of 86.71%.
The random state for the LR technique was set at 10, with a maximum of 1000 iterations. According to *Tables 7*, *9* and (*Figure 3*) the results demonstrate that it achieved accuracy rates of 79.59% and 77.64% over the test set for binary classification, making use of the complete and decreased feature spaces appropriately. With the number of neighbours selected as 3, 5, 7, 9, and 11, trained various models using the KNN algorithm. The outcomes reveal that in the multi-class classification configuration, a classifier using KNN with three neighbours achieved an 83.19% test outcome while using a complete feature space. Without overfitting, the KNN algorithm produced a test accuracy of 84.46% in the case of the decreasing feature dimension.

Regarding binary identification, the SVM approach with the radial basis function (RBF) achieved an accuracy of 60.89% and 62.42% using the complete and simplified input spaces, respectively. In comparison, the SVM used 42 and 19 features (*Figure 4*), respectively, to achieve test accuracies of 61.09% and 61.53% for multiclass detection.

For the DT classifier experimented with various models based on the tree's deepest depth. The following numbers could be assumed for the depth values: maximum depth values are equal to 2, 5, 7, 8, and 9. when it comes to binary classification setup, the outcomes indicate that the DT obtained an accuracy of 88.13% using 42 attributes, as opposed to a test outcome of 85.85% using 19 features. When conducting the binary-classification with the complete attribute size as well as the reduced one, the DT received a higher test accuracy rate in comparison to other ML approaches.

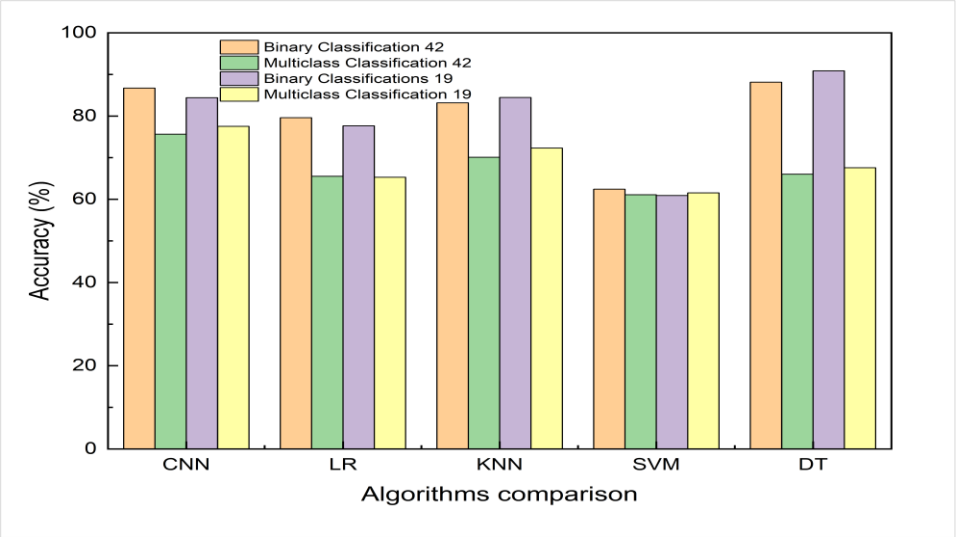

Figure 3 Classification of testing data accuracy

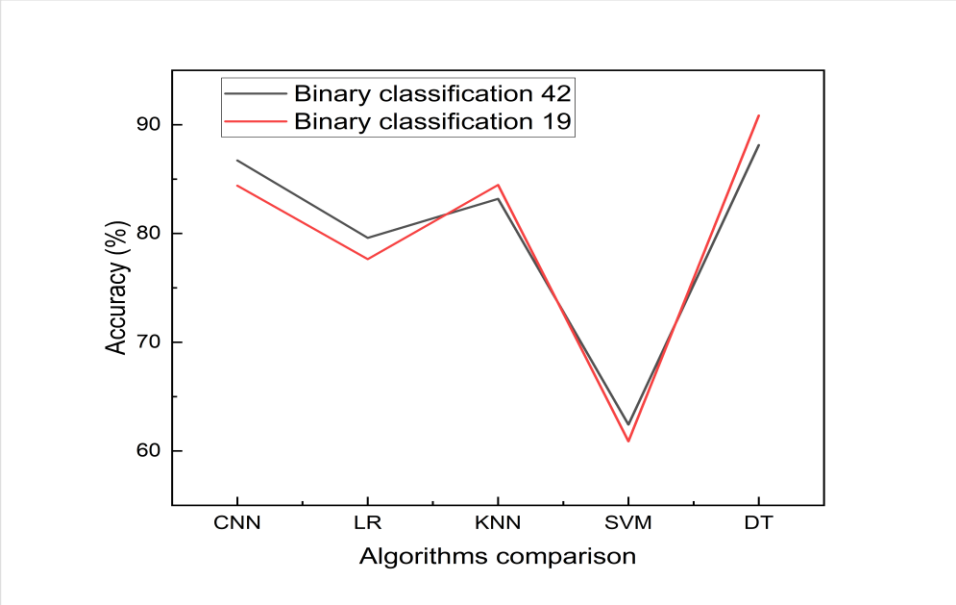

Figure 4 Comparison graph for binary classification

5.Discussion

After multiple testing iterations, the feature vector's dimension was reduced from 42 to 19, which helped the bulk of the ML approaches employed in this study increase their test accuracy level for the multiclass classification techniques. In the CNN scenario, from 75.62%, an increase of 42 characteristics with 30 nerves in the interred layer to 77.51% was noted for 19 attributes and 15 nerves in the interred layer. This research indicates that it is possible to reduce the variety of nodes needed in the CNN network by halving the complexity of the CNN approach by using only the most essential elements. This is also demonstrated by the trade-off between recall (77.53%) and precision (75.50%) in the test set. An F1-Score of 77.28% was obtained as a result similarly, the CNN technique out-performed other methods in terms of the recall, accuracy, and precision derived from outcome data. Accordingly, a confusion graph (*Figure 5*) regarding the CNN multiclass classifying method was constructed to assess the effectiveness of each attack class available in the dataset UNSW-NB15.

The test accuracy for the DT classifier improved from 66.03 to 67.57% when the multi-classification schemes 42 and 19 attributes were used, respectively. Furthermore, using the whole and decreased feature size of the UNSW-NB15, the DT improved test accuracy from 88.13 to 90.85%. The results showed that XGBoost aided in boosting the DT algorithm's capability for prediction.

The DT XGBoost algorithm significantly outperforms any other ML techniques for the binary method, while the CNN XGBoost algorithm outperforms all other ML methods in the case of the multiclass configuration, as shown in *Table 11*, when the approaches proposed in this research are compared to those discussed in the articles. *Table 11* shows the comparison of binary and multiclass classification using the different datasets.
Also, the XGBoost-DT got a test accuracy score of 90.85% compared to 81.42% for the GA-LR-DT with 20 UNSW-NB15 features, which is different from the outcomes in [17], where the GA-LR-DT was used. Moreover, the results obtained from this research outperform those obtained from previous research in which the sigmoid PIO selected 14 optimal features from the UNSW-NB15 and utilized the validation dataset to attain an accuracy score of 91.30%. *Table 9* illustrates the difference in validation accuracy between the XGBoost-artificial neural network (ANN), XGBoost-KNN, and XGBoost-DT, which were 93.66%, 94.73%, and 93.81%, respectively.

Limitations

This system's performance degrades with highly dimensional data. It is critical to implement a proper attribute extraction technique that can get rid of some attributes that have no significance for classification. Furthermore, models that are trained on incredibly unbalanced sets of data show low detection accuracy and a high false positive rate (FPR).

Table 11 Comparison table for binary classification and multiclass classification

Author	ML Method	Feature extraction technique	Dataset	Train Acc (%)	Binary Acc (%)	Multiclass Acc (%)
Kumar et al. [10]	DT	Instagram Algorithm	UNSW-NB15	-	-	57.01
Sekhar et al. [25]	J48	Particle Swarm Optimization, GA	NSL-KDD, UNSW-NB15	90.48	-	-
Sekhar et al. [25]	SVM	Particle Swarm Optimization, GA	NSL-KDD, UNSW-NB15	90.11	-	-
Saba et al. [43]	Bagging Forest	GA, Particle Swarm Optimization	BOT-IOT	91.27	-	-
Imran et al. [46]	RF	Instagram Algorithm	KDD CUP99	-	85.78	-
Wu et al. [32]	CNN-BiLSTM	Sequential Minimal Optimization	NSL-KDD	-	-	77.16
Proposed research	DT	XGBoost Algorithm	UNSW-NB15	94.12	90.85	67.57
	CNN	XGBoost Algorithm		94.21	84.39	77.51
	LR	XGBoost Algorithm		89.20	77.64	65.29
	KNN	XGBoost Algorithm		95.86	84.46	72.30
	SVM	XGBoost Algorithm		75.42	60.89	53.95

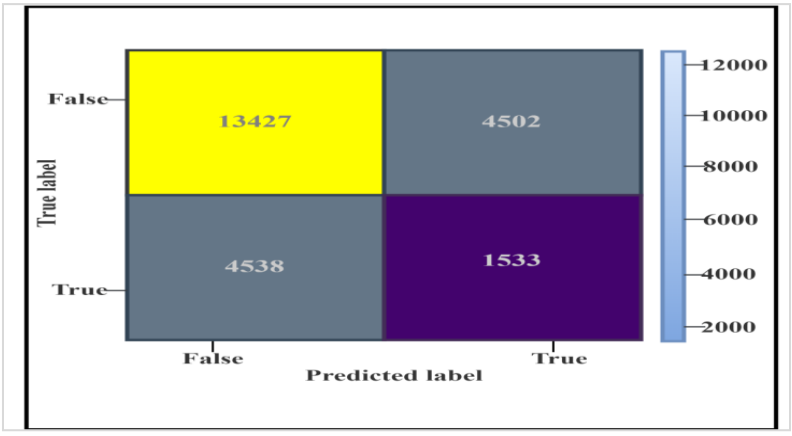

Figure 5 Confusion matrix

6.Conclusion and future work

This research investigated the use of the XGBoost-method for attribute selection combined with several ML techniques, such as CNN, DT, LR, KNN, and SVM to construct precise IDS. The efficiency of these methods was analyzed using the UNSW-NB15 network datasets. Both the multi and binary-classification attributes were taken into account in this article. Additionally, UNSW-NB15 was assigned to the XGBoost based attribute selection process, which resulted in the selection of 19 ideal attributes. In order to contextualize our findings, researchers carried out a thorough review of the literature and looked at several feature selection techniques used in the UNSW-NB15 network dataset. In addition, we generated a list of the efficiency outcomes achieved by the various classifiers used in the article and contrasted them with those attained by our suggested technique. The UNSW-NB15 dataset's whole feature space was used for our initial investigations, utilizing the suggested ML techniques. The trials were then conducted utilizing the XGBoost feature extraction approach that was reduced to create the vector feature in this work. The results of the experiment show that making use of a restricted desirable feature vector offers advantages in terms of decreasing method ambiguity and improving the deduction accuracy on validation data. The XGBoost-CNN is one significant illustration by reducing the number of nodes in the standalone CNN by 50%, it was able to lower the variety of neurons in the concealed layer. Similarly, the XGBoost-DT and the XGBoost-KNN have both experienced an enhancement in their efficiency when compared to test data. Furthermore, an examination of the effectiveness of particular training inside the UNSW-NB15 network dataset employing the XGBoost-CNN method revealed that this approach outperforms itself when forecasting the majority-classes and performs below itself when forecasting the minority classes. In future research, we plan to use an ensemble learning technique to improve the minority class representation in the UNSW-NB15 training dataset.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

The dataset used in this study is publicly available at https://research.unsw.edu.au/projects/unsw-nb15-dataset.

Author's contributions statement

S.V. Sugin: Conceptualization, methodology, dataset collection, result analysis, implementation, Writing – review and editing. **M. Kanchana**: Supervision, Writing – review and editing.

References

[1] Keserwani PK, Govil MC, Pilli ES, Govil P. A smart anomaly-based intrusion detection system for the internet of things (IoT) network using GWO–PSO–RF model. Journal of Reliable Intelligent Environments. 2021; 7:3-21.

[2] Maseer ZK, Yusof R, Bahaman N, Mostafa SA, Foozy CF. Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. IEEE Access. 2021; 9:22351-70.

[3] Le JL, Goedeme T, Mentens N. Machine learning for misuse-based network intrusion detection: overview, unified evaluation and feature choice comparison framework. IEEE Access. 2021; 9:63995-4015.

[4] Rizvi S, Scanlon M, Mcgibney J, Sheppard J. Deep learning based network intrusion detection system for resource-constrained environments. In international conference on digital forensics and cyber crime 2022 (pp. 355-67). Cham: Springer Nature Switzerland.

[5] Kim T, Pak W. Robust network intrusion detection system based on machine-learning with early classification. IEEE Access. 2022; 10:10754-67.

[6] Maddu M, Rao YN. Network intrusion detection and mitigation in SDN using deep learning models. International Journal of Information Security. 2023:1-4.

[7] Brindha DV, Ranjan NM, Sharma H. IoT attack detection and mitigation with optimized deep learning techniques. Cybernetics and Systems. 2022:1-27.

[8] Bashah NS, Simbas TS, Janom N, Aris SR. Proactive DDoS attack detection in software-defined networks with snort rule-based algorithms. International Journal of Advanced Technology and Engineering Exploration. 2023; 10(105):962-89.

[9] Chikkalwar SR, Garapati Y. Network intrusion detection system using bacterial foraging optimization with random forest. International Journal of Advanced Technology and Engineering Exploration. 2023; 10(105):1037-49.

[10] Kumar V, Sinha D, Das AK, Pandey SC, Goswami RT. An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset. Cluster Computing. 2020; 23:1397-418.

[11] Horak T, Strelec P, Huraj L, Tanuska P, Vaclavova A, Kebisek M. The vulnerability of the production line using industrial IoT systems under DDOS attack. Electronics. 2021; 10(4):1-31.

[12] Ravi V, Chaganti R, Alazab M. Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. Computers and Electrical Engineering. 2022; 102:108156.

[13] Al S, Dener M. STL-HDL: a new hybrid network intrusion detection system for imbalanced dataset on big data environment. Computers & Security. 2021; 110:102435.
[14] Riyad AM. A DDoS defence framework in software defined network using ensemble classifier with rough set theory based feature selection. International Journal of Advanced Technology and Engineering Exploration. 2021; 8(82):1120-35.
[15] Ali M, Haque MU, Durad MH, Usman A, Mohsin SM, Mujlid H, et al. Effective network intrusion detection using stacking-based ensemble approach. International Journal of Information Security. 2023; 22(6):1781-98.
[16] Alazzam H, Sharieh A, Sabri KE. A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert Systems with Applications. 2020; 148:113249.
[17] Kamalakkannan D, Menaga D, Shobana S, Daya SKV, Rajagopal R, Tiwari M. A detection of intrusions based on deep learning. Cybernetics and Systems. 2023:1-5.
[18] Albasheer H, Md SM, Mubarakali A, Elsier TO, Salih S, Hamdan M, et al. Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: a survey. Sensors. 2022; 22(4):1-15.
[19] Wang Z, Jiang D, Huo L, Yang W. An efficient network intrusion detection approach based on deep learning. Wireless Networks. 2021:1-4.
[20] Nazir S, Patel S, Patel D. Autoencoder based anomaly detection for SCADA networks. International Journal of Artificial Intelligence and Machine Learning. 2021; 11(2):83-99.
[21] Laskar MT, Huang JX, Smetana V, Stewart C, Pouw K, An A, et al. Extending isolation forest for anomaly detection in big data via K-means. ACM Transactions on Cyber-Physical Systems. 2021; 5(4):1-26.
[22] Ortega-fernandez I, Sestelo M, Burguillo JC, Piñón-blanco C. Network intrusion detection system for DDoS attacks in ICS using deep autoencoders. Wireless Networks. 2023:1-7.
[23] Sah G, Banerjee S, Singh S. Intrusion detection system over real-time data traffic using machine learning methods with feature selection approaches. International Journal of Information Security. 2023; 22(1):1-27.
[24] Ogundokun RO, Awotunde JB, Sadiku P, Adeniyi EA, Abiodun M, Dauda OI. An enhanced intrusion detection system using particle swarm optimization feature extraction technique. Procedia Computer Science. 2021; 193:504-12.
[25] Sekhar R, Sasirekha K, Raja PS, Thangavel K. A novel GPU based intrusion detection system using deep autoencoder with fruitfly optimization. SN Applied Sciences. 2021; 3(6):1-16.
[26] Ahmadi AF, Milani FA, Khanchi S. Hybrid machine learning-based approaches for feature and overfitting reduction to model intrusion patterns. Journal of Cybersecurity and Privacy. 2023; 3(3):544-57.
[27] Sugin SV, Kanchana M. Machine learning-based intrusion detection of imbalanced traffic on the network: a review. In the international conference on recent innovations in computing 2022 (pp. 741-53). Singapore: Springer Nature Singapore.
[28] Ghani H, Salekzamankhani S, Virdee B. A hybrid dimensionality reduction for network intrusion detection. Journal of Cybersecurity and Privacy. 2023; 3(4):830-43.
[29] Soumya TR, Revathy S. A novel approach for cyber threat detection based on angle-based subspace anomaly detection. Cybernetics and Systems. 2022:1-10.
[30] Rani M, Gagandeep. Effective network intrusion detection by addressing class imbalance with deep neural networks multimedia tools and applications. Multimedia Tools and Applications. 2022; 81(6):8499-518.
[31] Kanchana M. Detection of traffic on the network based on a real dataset for the IIM method and ML-TSDS algorithm. In international conference on automation, computing and renewable systems 2022 (pp. 614-22). IEEE.
[32] Wu T, Fan H, Zhu H, You C, Zhou H, Huang X. Intrusion detection system combined enhanced random forest with SMOTE algorithm. EURASIP Journal on Advances in Signal Processing. 2022; 2022(1):1-20.
[33] Kasongo SM, Sun Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. Journal of Big Data. 2020; 7(1):1-20.
[34] Gu J, Lu S. An effective intrusion detection approach using SVM with naïve bayes feature embedding. Computers & Security. 2021; 103:102158.
[35] Asha VS, Ganesh RK. An AI based IDS framework for detecting DDoS attacks in cloud environment. Information Security Journal: A Global Perspective. 2023:1-3.
[36] Sugin SV, Kanchana M. Improved cyber attack detection using MLB-FQS: a novel modified lagrange butterfly-based fuzzy quasi-linear SVM algorithm. IETE Journal of Research. 2023:1-5.
[37] Sommestad T, Holm H, Steinvall D. Variables influencing the effectiveness of signature-based network intrusion detection systems. Information Security Journal: a Global Perspective. 2022; 31(6):711-28.
[38] Fu Y, Du Y, Cao Z, Li Q, Xiang W. A deep learning model for network intrusion detection with imbalanced data. Electronics. 2022; 11(6):1-13.
[39] Le TT, Oktian YE, Kim H. XGBoost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems. Sustainability. 2022; 14(14):1-21.
[40] Bagui S, Li K. Resampling imbalanced data for network intrusion detection datasets. Journal of Big Data. 2021; 8(1):1-41.
[41] Chew YJ, Lee N, Ooi SY, Wong KS, Pang YH. Benchmarking full version of GureKDDCup, UNSW-

NB15, and CIDDS-001 NIDS datasets using rolling-origin resampling. Information Security Journal: A Global Perspective. 2022; 31(5):544-65.
[42] Lee J, Park K. GAN-based imbalanced data intrusion detection system. Personal and Ubiquitous Computing. 2021; 25(1):121-8.
[43] Saba T, Rehman A, Sadad T, Kolivand H, Bahaj SA. Anomaly-based intrusion detection system for IoT networks through deep learning model. Computers and Electrical Engineering. 2022; 99:107810.
[44] Le KH, Nguyen MH, Tran TD, Tran ND. IMIDS: an intelligent intrusion detection system against cyber threats in IoT. Electronics. 2022; 11(4):1-16.
[45] Yu J, Ye X, Li H. A high precision intrusion detection system for network security communication based on multi-scale convolutional neural network. Future Generation Computer Systems. 2022; 129:399-406.
[46] 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.

Mr. S.V. Sugin received his M.E Degree in Computer Science and Engineering from Anna University, Chennai in 2014. From December 2021, he has been doing Full Time Ph.D. in Department of Computing Technologies, School of Computing, of SRM Institute of Science and Technology, Kattankulathur Campus, Chennai. His research area includes Artificial Intelligence, Machine Learning and Network Security.
Email: sugin.sv@gmail.com

Dr. M. Kanchana received her Ph.D. Degree in Computer Science and Engineering from Anna University, in 2018. From June 2019, she has been working as an Associate Professor in Department of Computing Technologies, School of Computing, of SRM Institute of Science and Technology, KTR Campus, Chennai. She published more than 25 papers in SCI and Scopus indexed journals. She also published in 5 patents and 5 books. Her research area includes Artificial Intelligence, Deep Learning, Machine Learning, Image Processing, Data mining, and Pattern Recognition.
Email: kanchanm@srmist.edu.in

Appendix I

S. No.	Abbreviation	Description
1	ANN	Artificial Neural Network
2	bi-LSTM	bi-Directional Long Short Term Memory
3	CNN	Convolutional Neural Network
4	CN	Computer Network
5	CSV	Comma Separated Value
6	DDoS	Distributed Denial of Service
7	DIDS	Distributed IDS
8	DT	Decision Tree
9	DR	Detection Rate
10	EM	Euclidean Metric
11	FAR	False Alert Rate
12	FMI	Flexible Mutual Information
13	FN	False Negative
14	FM	F-Measure
15	FP	False Positive
16	FPR	False Positive Rate
17	GA	Genetic Algorithm
18	HIDS	Host-based IDS
19	HYIDS	Hybrid IDS
20	IDS	Intrusion Detection Systems
21	IG	Information Gain
22	IPS	Intrusion Prevention Systems
23	KNN	K-Nearest Neighbour
24	LR	Logistic Regression
25	LS-SVM	Least Square SVM
26	ML	Machine Learning
27	MLP	Multilayer Perceptrons
28	NIDS	Network based IDS
29	NSL-KDD	Network Security Dataset
30	OSS	One Side Selection
31	PIO	Pigeon Inspired Optimizer
32	RBF	Radial Basis Function
33	ReLU	Rectified Linear Unit
34	RF	Random Forest
35	R2L	Remote to Local
36	SE	Sigmoid Expression
37	SVM	Support Vector Machine
38	SGD	Stochastic Gradient Descent
39	SMOS	Synthetic Minority Over Sampling
40	TN	True Negative
41	TP	True Positive
42	TS	Two Stage
43	UNSW-NB15	Network Dataset
44	U2R	User to Root
45	XGBoost	Extreme Gradient Boosting