Journal of Big Data

RESEARCH Open Access

A hybrid machine learning method for increasing the performance of network intrusion detection systems

Achmad Akbar Megantara and Tohari Ahmad*

*Correspondence:
tohari@if.its.ac.id
Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Abstract

The internet has grown enormously for many years. It is not just connecting computer networks but also a group of devices worldwide involving big data. The internet provides an opportunity to make various innovations for any sector, such as education, health, public facility, financial technology, and digital commerce. Despite its advantages, the internet may contain dangerous activities and cyber-attacks that may happen to anyone connected through the internet. To detect any cyber-attack intrudes on the network system, an intrusion detection system (IDS) is applied, which can identify those incoming attacks. The intrusion detection system works in two mechanisms: signature-based detection and anomaly-based detection. In anomaly-based detection, the quality of the machine learning model obtained is influenced by the data training process. The biggest challenge of machine learning methods is how to build an appropriate model to represent the dataset. This research proposes a hybrid machine learning method by combining the feature selection method, representing the supervised learning and data reduction method as the unsupervised learning to build an appropriate model. It works by selecting relevant and significant features using feature importance decision tree-based method with recursive feature elimination and detecting anomaly/outlier data using the Local Outlier Factor (LOF) method. The experimental results show that the proposed method achieves the highest accuracy in detecting R2L (i.e., 99.89%) and keeps higher for other attack types than most other research in the NSL-KDD dataset. Therefore, it has a more stable performance than the others. More challenges are experienced in the UNSW-NB15 dataset with binary classes.

Keywords: Intrusion detection system, Feature selection, Data reduction, Decision tree, Local Outlier Factor, Network security, Network infrastructure

Introduction

Since the ARPANET first introduced the internet in 1969, it has grown significantly that many devices are connected for transferring various data [1]. These include network servers, portable computers (notebooks), and mobile devices, which may connect to the cloud environment containing big data. This condition provides an opportunity to make various innovations in any sector, such as financial technology, health, digital commerce, education, and public facility.

Besides various advantages and opportunities that the internet can provide, various activities threaten users' security and privacy, for example, Denial-of-Service (DoS), phishing, Man-in-the-Middle (MitM), malware, password attacks, backdoors, and rootkits. These attacks can cause harmful activities like losing some of our most valuable assets, including password accounts, financial information, user privacy, business plans, and other sensitive data [2].

To prevent the network system from cyber-attacks, some industries and companies implement network intrusion detection systems (IDS) to identify and mitigate incoming attacks in their network system [3]. The intrusion detection system's primary benefit is to ensure the network administrator is accurately informed of a dangerous activity. The intrusion detection system monitors and identifies network traffic data and triggers alerts when suspicious activity or identified threats are detected, so the network administrator can examine the activity and take the appropriate decision [4].

The intrusion detection system works in two mechanisms: signature-based detection and anomaly-based detection [5]. Signature-based detection uses a known list of rules or indicators from the system attack database to specify whether the activity is malicious or not, while anomaly-based detection identifies the attack based on unusual user behavior patterns [6]. If there are users who perform unusual actions or activities, it can be detected as an attack.

In anomaly-based detection IDS, the use of behavior offers the best performance to identify the attack activity by combining it with several data mining and machine learning methods [7]. These methods intelligently identify and provide a new perspective of today's attack types across the global computer networks. However, the use of the machine learning method in IDS still suffers from several problems. The biggest challenge of machine learning methods is how to build an appropriate model to represent the dataset [8].

The quality of the machine learning model obtained is influenced by the data training process. Good training data can be generated by performing data pre-processing steps such as feature selection and data reduction. Feature selection is the process of selecting which features will be used based on their significant value to the data label of each feature calculated [9]; data reduction is the process for dropping the records/instances that deviate from other data, called outliers data [10].

The feature selection process is divided into two types: filter and wrapper methods. The filter method is a feature selection process, which calculates each feature's significant value using various statistical algorithms for their correlation and relevance to the dataset label. The wrapper method calculates the significant value of each feature by evaluating of subset generated for each iteration. The filter method-based feature selection challenge is in how to determine the threshold for the significant value generated by the filter method. In some previous research, the threshold value is usually manually configured and tested by exploring each possible value. The problem with this technique is that it is user-dependent that the performance relies on the user selection of the appropriate threshold.

The data reduction process is divided into two focuses: global outlier and local outlier. The global outlier is calculated from the whole data in the set, while the local outlier is calculated from specific data in the entire dataset [11]. The outlier detection problem

is how to determine the parameter value to represent whether the value is an outlier or not.

This research proposes a hybrid machine learning method by combining feature selection using feature importance ranking from the Decision Tree Algorithm with data reduction techniques using Local Outlier Factor (LOF) to increase the network intrusion detection system's performance. We also propose a technique to determine the threshold value in the feature selection process and identify the outlier data in the data reduction process.

This paper is divided into five parts. The first is the introduction, which introduces the background and the general problem statement of this research. The second part is related work, consisting of several research pieces related to this research. The proposed method is explained in the third part. The experimental result with the performance comparison with that of other research is discussed in the fourth part. The last is the conclusion of this study.

Related work

Some research was previously released on machine learning methods in the use of network intrusion detection systems. Each research has different topic concerns, such as feature selection, data reduction, and classification method optimization.

The most common challenges in a network dataset are the complexity and various types of features in that dataset. Various methods have been implemented to deal with these challenges by selecting features in the dataset. Eid et al. [12] use linear correlation-based feature selection for selecting features by calculating the similarity between two random variables. This proposed method reduces the number of features in the NSL-KDD [13] dataset from 41 to 17 features. Amiri et al. [14] propose a modified mutual information method by calculating the maximum relevancy and minimum redundancy as the parameter to evaluate each feature. The mutual information method works by evaluating the arbitrary dependency between two variables to generate mutual information coefficients. This proposed method has shown the crucial features of each class in the KDD dataset. Mohammed and Ahmed [15] applied ANOVA-PCA to select features in the dataset by combining the ANOVA method that analyzes each data variance and PCA that computes each feature's principal component. Their proposed method can generate a significant value of each feature for both the KDD and NSL-KDD datasets.

Unlike filter-based feature selection, Almasoudy et al. [16] present the wrapper method using Extreme Learning Machine (ELM) to calculate each subset generated in the NSL-KDD dataset. Zhou et al. [17] propose an ensemble-based scheme by combining filter-based and wrapper-based methods in feature selection. They use correlation-based feature selection to obtain the features and Bat Algorithm (BA) to find the dataset's best subset. The other ensemble-based method, proposed by Aljawarneh et al. [18], uses Information Gain (IG) to calculate each feature's significant value, and several methods are applied to calculate possible subsets in the dataset. Research from Nkiama et al. [19] takes ANOVA F-test to calculate the importance of the dataset's features based on each data variance. The number of features used is generated, and the selected features are processed by a wrapper method using the

Recursive Feature Elimination (RFE). The experimental results show that the proposed scheme can increase the system's performance, especially accuracy scores.

From several studies in the feature selection process, the ensemble-based method has become a popular method for the feature selection process because it can solve both filter-based and wrapper-based methods' problems. The filter-based method calculates each feature's significant value and the wrapper-based method for evaluating the generated subset. However, the problem in the ensemble-based method is how to configure the threshold in the filter-based method, wherein in the previous research [16–18], it was configured manually by the researcher. Because of this manual configuration, the threshold will depend on the value given by the user, which is likely different for every setup. Therefore, the threshold calculation mechanism proposed in this research contributes to avoiding the user-dependent in its configuration.

The problem with large amounts of datasets is the existence of outlier and redundant data in the set. In 2020, Iman and Ahmad [20] developed a method for optimizing feature selection using a data reduction process. They proposed a cluster size of the *k*-means clustering by calculating the minimum, maximum, and median clusters. Data inside the cluster are for the classification process, increasing the method's accuracy even though it still suffers from several problems such as irrelevant features and biased data. Prasad et al. [21] present a new proposed modified *k*-means clustering to detect outliers and redundant data by computing semi-identical sets and creating a number of micro-clusters in the KDD dataset. Pu et al. [22] propose a method called SSC-OCSVM by combining sub-space clustering with a one-class support vector machine to detect malicious activity in the NSL-KDD dataset. In 2017, Saleh et al. [23] introduced Hybrid Intrusion Detection System (HIDS) by combining four modules: data pre-processing module, NBFS for feature selection, outlier detection using OSVM, and PKNN for taking the decision. In 2021, Gupta et al. [24] proposed a method called NoC Efficiency Through Supervised Machine Learning (EE-NOSML) to optimize the energy efficiency of Wireless Sensor Network by creating the neighborhood search calculations.

In the data reduction process, clustering has become one of the most used methods for detecting the outlier data in the dataset; it is also used to calculate the distance between data. In 2021, Gupta et al. [25] developed several clustering methods to optimize the performance of Wireless Sensor Network routing protocols by finding the optimal path for data packets from source to destination. Various clustering methods are applied to generate the cluster, and different calculation mechanisms of cluster size and outlier threshold value have been proposed in several research studies. In the previous research, determining what data reduction method should be used, how to configure the cluster size, and how to set the threshold value for distinguishing the outlier data are the primary concerns for optimizing the IDS performance. Therefore, in this research, a mechanism for calculating an outlier value threshold is proposed to set the limit of outlier data in the IDS dataset.

For evaluating the methods, previous research took KDD CUP 99, which was developed as a standard of network intrusion dataset, and was becoming a part of the KDD Archive in the UCI Machine Learning Data Repository [26]. The following research uses the new version of the KDD CUP 99 [13] and UNSW-NB15 [27] datasets.

Proposed method

This research is primarily developed based on the previous research done by Megantara and Ahmad [28], which focuses on the feature selection process. This previous study proposed a feature importance ranking based on the decision tree method for selecting features and wrapper methods using recursive feature elimination to calculate the subset score. In this research, we propose a scheme called Hybrid Machine Learning Method, which combines both feature selection and data reduction processes. The detailed explanation of this research's proposed method can be described as follows (see Fig. 1).

- In the feature selection process, the importance value to the dataset label of each feature is calculated. We introduce a threshold mechanism by removing features with zero values to separate high importance features from low importance features and divide the rest with median data.
- In the data reduction process, we use the local outlier factor for detecting outliers for each data point. Normal/Gaussian distribution will be used to configure the cut-off value for the anomaly score.

Feature importance ranking

The purpose of the feature selection process is only to take significantly relevant features with the label decision. Feature selection is divided into three mechanisms: Filter-based, wrapper-based, and embedded-based methods. The filter-based method first selects features using various statistical methods or algorithms to calculate each feature's important/significant value. Differently, the wrapper-based method chooses the dataset's features by finding the best possible subset combination. In this research, we explore the embedded-based method by combining both filter-based and wrapper-based methods. The dataset's significant/essential features are generated by calculating the probability

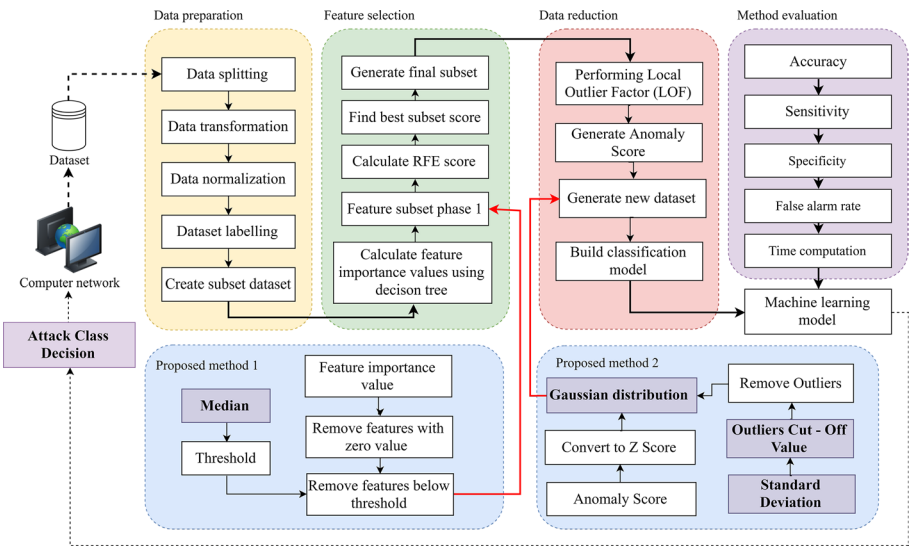

Fig. 1 The flow diagram of the research comprises four parts: Data preparation, Feature selection, Data reduction and Method evaluation

number of the data for each feature in the dataset. It is to reach the decision node of the decision tree-based method [29]. Here, Eq. (1) [29] is applied to calculate each node's importance value for every feature in the dataset. In this formula, ni_i is the importance values of each node j, w_j is the weighted value of each instance in node j, C_j is the impurities value of node j and j, $left(j)$ and $right(j)$ is the child nodes.

$$ni_i = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \tag{1}$$

Since each node's importance values are generated, each feature's important/significant value can be calculated using Eq. (2) [29]. Here, fi_i is the importance/significance of each feature, and ni_i is the importance value of each node.

$$fi_i = \frac{\sum_{j:node\,j\,split\,on\,feature} ni_j}{\sum_{k \in all\,nodes} ni_k} \tag{2}$$

After each feature's importance values are obtained, a threshold for separating selected features from those not selected in the dataset is configured. For this purpose, we propose a method whose mechanism is given in Fig. 2.

In the previous research, especially most filter-based method feature selection, the threshold value is manually configured. It can be done by testing every possible generated value. Those methods still suffer from several problems, mainly because this value is user-dependent. So, in this research, we present a mechanism to solve these problems. Firstly, the importance values of each feature will be sorted from the highest to the lowest values. Features with zero importance value mean that they are irrelevant to the label decision; therefore, those features will be removed first. Secondly, the rest of the features will be classified into two groups with a median value as the threshold. This value in the data set can be calculated using Eq. (3) for the odd and Eq. (4) for the even number of data.

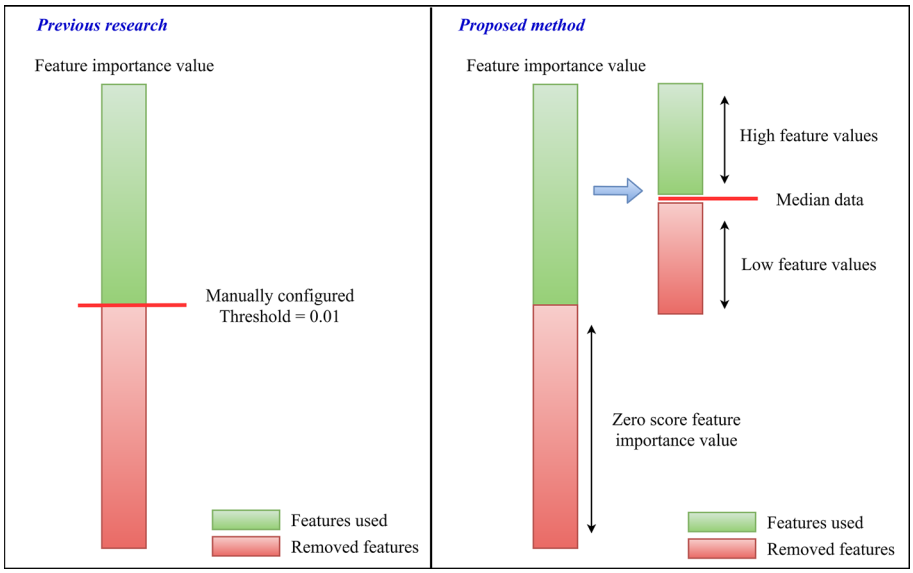

Fig. 2 The method comparison between previous research [28] with proposed method in this research

$$Me = X_{\frac{(n+1)}{2}} \tag{3}$$

$$Me = \frac{X_{\frac{n}{2}} + X_{\frac{(n+1)}{2}}}{2} \tag{4}$$

Here, X is the middle value, and n is the total number of data.

Local Outlier Factor

The Local Outlier Factor (LOF) is to calculate the local outlier for each data point. The LOF process produces a score known as an anomaly score, representing how far each data points to its neighbors. The higher the anomaly score in the data point, the far the data point from other data, which is called anomaly data [30].

The LOF method is divided into four steps: Determining the *k*-value to initiate the cluster size, calculating the reachability distance for each data point, calculating the local reachability distance value, and generating the LOF score/anomaly score for each data point. The illustration of the LOF method is provided in Fig. 3. It is defined that the *k*-value is that to initiate the size of a cluster. This value will affect how anomaly data can be generated.

Reachability distance (RD) is the distance from each data point to its maximum distance of *k*-value data points. The function of the reachability distance is to find the perimeter area for each data points for mapping the number of other nearest data points. If there are many other data points inside the perimeter area, it means that the data are not outliers. This value can be calculated using Eq. (5) [30], where *RD* is the reachability distance value of each data point, K is the *k*-value data points, X_A is the data points, and $X_{A'}$ is the furthest distance data point.

$$\mathrm{RD}(X_A, X_{A'}) = \max(K - \mathrm{distance}(X_{A'}), \mathrm{distance}(X_A, X_{A'})) \tag{5}$$

After the RD value is obtained, the Local Reachability Distance (LRD) value is calculated using Eq. (6) [30] to determine the distance ratio for every nearest neighbor inside the cluster, where *LRD* is the local reachability distance value and *Nk(A)* is the *K*-Neighbors.

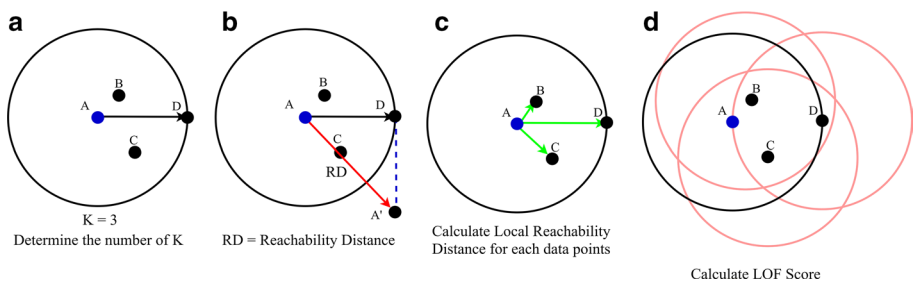

Fig. 3 The illustration of Local Outlier Factor (LOF) process. **A** Determine the number of k-value **B** calculate the Reachability Distance (RD) **C** Calculate the Local Reachability Distance (LRD) **D** Calculate the LOF Score for each data points

$$LRD_k(A) = \frac{1}{\sum_{X_j \in N_k(A)} \frac{RD(A,X_j)}{|N_k(A)|}} \tag{6}$$

The LRD value is to calculate the anomaly score/LOF score for each data point. The LOF score is the ratio between the LRD value of each data point and all the data points. It is used for comparing the distance ratio between each data point and the other data points. To calculate the anomaly score/LOF score, Eq. (7) [30] is applied, where *LOF* is the LOF score/anomaly score, *LRD* is the local reachability distance values, and *Nk(A)* is the *K*-Neighbours.

$$LOF_k(A) = \frac{\sum_{X_j \in N_k(A)} LRD_k(X_j)}{|N_k(A)|} \frac{1}{LRD_k(A)} \tag{7}$$

The generated LOF score is then distributed using a standard normal distribution (Gaussian distribution) to configure the cut-off value between normal and anomaly data. First, the LOF score is converted into *Z*-score to determine how far each LOF score is from the mean data in the dataset. For converting the LOF score to the *Z-score*, Eqs. (8–10) are performed.

$$m = \frac{1}{n} \sum_{i}^{n} X_i \tag{8}$$

$$std = \sqrt{\frac{\sum_{i=1}^{n} (X_i - m)^2}{n_i}} \tag{9}$$

$$Z = \frac{(X - m)}{std} \tag{10}$$

Equations (8–10) are the formula for converting the LOF score/anomaly score into the *Z*-score, where *m* is the mean value, *std* is the standard deviation, *z* is the *Z*-score value, *n* is the total data number in the dataset, and *X* is the LOF score/anomaly score.

After the *Z*-score of each data point is generated, they are distributed to the standard normal distribution. In this standard normal distribution, the *x*-axis is the data point values, and the *y*-axis is the *Z*-score values. The standard deviation is calculated as the cut-off value for generating the data subsets. The mean score from the set of data is the maximum score of data distribution. Standard deviation 3 means that 99.73% of data will be used while the rest will be removed, while standard deviation 2 means that 95.45% of data will be used while the rest will be removed. Furthermore, standard deviation 1 represents that 68.27% of data will be used, and the rest will be removed. These cut-off values depict how far each LOF score/anomaly score is to its maximum score.

Experimental results

NSL-KDD and UNSW-NB15 datasets

Two datasets are taken for testing the proposed method: The NSL-KDD [13] and UNSW-NB15 [27]. This first dataset is the latest version of the KDD CUP 99 dataset, which is introduced with no redundant data, duplicated data, and proportionally

distributed for training and testing data. It is obtained by generating the new data from KDD Cup 99. The second dataset is UNSW-NB15, the latest IDS dataset introduced by the University of New South Wales at the Australian Defence Force Academy. The UNSW-NB15 dataset is generated by creating a synthetic environment configuration with virtual servers made by IXIA traffic generator. Several scenarios are applied, and the traffic data are collected to generate the dataset. This set consists of flow, basic, content, time, and some additional features. Both NSL-KDD and UNSW-NB15 datasets consist of categorical, numerical, and binary features. The data distribution of each class is presented in Table 1.

Classification

The decision tree classifier is taken as the machine learning model to classify the dataset's training data. The first reason why this classifier is performed in this research is that the base method of the feature selection process in the feature importance ranking method is also the decision tree. Secondly, the condition of the generated dataset from the proposed method is suitable enough for the decision tree method's characteristics. It includes the minimum number of features and data in the dataset, the heterogeneity of each feature's data in the dataset, and the inexistence of the duplicate or redundant data from the newly generated dataset.

Method evaluation

There are five parameters for evaluating whether the proposed method can solve the existing problem or not. These parameters are accuracy, sensitivity, specificity, false alarm rate, and computational time. The accuracy, sensitivity, specificity, and false alarm rate are calculated using Eqs. (11–14). The computation time is generated by obtaining computation time from the first to the last method. In this research, our specification hardware is 8 Gb RAM, i5-3210 M CPU 2.5 GHz, NVIDIA GeForce GT 630 M, and Jupyter Notebook with Python 3.7.7.

i. Confusion matrix
   Table 2 is the confusion matrix that is used in this research. It consists of 2 predicted classes and 2 actual classes. Class 0 refers to normal conditions, and class 1 refers to attack activity in the UNSW-NB15 dataset and DoS/Probe/R2L/U2R activity for the NSL-KDD dataset. True Positive (TP) is an attack activity that

Table 1 Data distribution of each dataset

Dataset	Class	Training	Testing
NSL-KDD [13]	Normal	67,343	9711
	DoS	45,927	7460
	Probe	11,656	2885
	R2L	995	2421
	U2R	52	67
UNSW-NB15 [27]	Normal	119,341	45,332
	Attack	56,000	37,000

Table 2 Confusion matrix

Predicted	Actual	
	0	**1**
0	TN	FN
1	FP	TP

is correctly predicted as an attack; False Positive (FP) is a normal activity that is incorrectly predicted as an attack; True Negative (TN) is a normal activity that is correctly predicted as normal; False Negative (FN) is an attack that is incorrectly predicted as normal.

ii. Accuracy
Accuracy represents the system's ability to correctly detect whether the activity is an attack or normal activity. The accuracy value can be calculated using the formula in Eq. (11).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

iii. Sensitivity
Sensitivity depicts the system's ability to detect the incoming activity as the actual attack among all detected attacks. The sensitivity value can be calculated using Eq. (12).

$$Sensitivity = \frac{TP}{TP + FN} \tag{12}$$

iv. Specificity
Contrary to sensitivity, specificity shows the system's ability to detect which incoming activity is the real normal activity among all detected normal data. It is calculated using Eq. (13).

$$specificity = \frac{TN}{TN + FP} \tag{13}$$

v. False alarm rate
False alarm rate is to find the amount of attack that is incorrectly predicted as the normal activity. The larger of false alarm rate value, the more attack activity was predicted as normal activity. For this purpose, Eq. (14) is implemented.

$$False\ Alarm\ Rate = \frac{FP}{FP + TN} \tag{14}$$

Feature selection process

In the feature selection process, the filter-based method and wrapper-based method are applied. The proposed method mechanism is performed whose experimental results are provided in Table 3. The features in the dataset are determined using a filter-based

Table 3 Selected features of NSL-KDD [13] and UNSW-NB15 [27] datasets

Dataset	Class	Filter method	Wrapper method	Selected features
NSL-KDD [13]	DoS	14	8	'src_bytes', 'dst_bytes', 'wrong_fragment', 'num_compromised', 'same_srv_rate', 'dst_host_serror_rate', 'dst_host_srv_serror_rate', 'service_ecr_i'
	Probe	21	14	'src_bytes', 'dst_bytes', 'count', 'rerror_rate', 'dst_host_same_srv_rate', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_rerror_rate', 'service_finger', 'service_ftp_data', 'service_http', 'service_private', 'service_smtp', 'service_telnet'
	R2L	15	14	'duration', 'src_bytes', 'dst_bytes', 'hot', 'num_failed_logins', 'num_root', 'num_access_files', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'service_ftp_data', 'service_imap4'
	U2R	14	4	'src_bytes', 'dst_bytes', 'hot', 'root_shell'
NSL-KDD [13]	Attack–normal	31	4	'src_bytes', 'dst_host_same_srv_rate', 'service_ecr_i', 'service_http'
UNSW-NB15 [27]	Attack–normal	25	11	'sbytes', 'sttl', 'sload', 'sinpkt', 'dinpkt', 'stcpb', 'tcprtt', 'synack', 'smean', 'ct_srv_src', 'ct_srv_dst'

method, which is a decision tree-based algorithm. The features generated from the NSL-KDD dataset with filter method for DoS, Probe, R2L, and U2R are 14, 21, 15, and 14, respectively. The attack–normal class for the UNSW-NB15 dataset generates 25 features from the filter method and 31 features from NSL-KDD.

Generated features from the filter method are used for the wrapper method to subset evaluation. It calculates the score of possible subset features combination. The best subset score for DoS, Probe, R2L, and U2R in the NSL-KDD dataset is 8, 14, 14, and 4 features, respectively. In comparison, the best subset score for the UNSW-NB15 dataset is 11 features in the subset method and 4 features in the NSL-KDD attack–normal class.

From the feature selection process, the selected features are obtained. These selected features represent how significant/relevant every feature in the dataset is to the decision label.

Data reduction process

In the data reduction process, local outlier/anomaly data are detected using the Local Outlier Factor (LOF) method. The standard normal distribution (Gaussian distribution) is performed to detect the outlier based on the standard deviation cut-off. The experimental results from local outlier detection are presented in Tables 4, 5. Those tables compare the total amount of data before the local outlier detection method is applied and the total amount of outlier data detected, respectively.

Performance evaluation

i. Accuracy

Figure 4 shows the accuracy score comparison between each step in the experiment. This proposed method shows the increasing accuracy score from using the

Table 4 Total data comparison between before and after data reduction

Dataset	Class	Total data before data reduction	Total data after data reduction		
			std = 1	std = 2	std = 3
NSL-KDD [13]	DoS	113,270	110,984	111,229	111,819
	Probe	78,999	78,937	78,937	78,938
	R2L	68,338	68,257	68,268	68,278
	U2R	67,395	67,382	67,382	67,382
NSL-KDD [13]	Attack-Normal	125,973	125,827	125,827	125,827
UNSW-NB15 [27]	Attack-Normal	175,341	169,599	172,543	173,880

Table 5 Outliers in the dataset

Dataset	Class	Total data after data reduction		
		std = 1	std = 2	std = 3
NSL-KDD [13]	DoS	2286	2041	1451
	Probe	62	62	61
	R2L	81	70	60
	U2R	13	13	13
NSL-KDD [13]	Attack–normal	146	146	146
UNSW-NB15 [27]	Attack–normal	5742	2798	1461

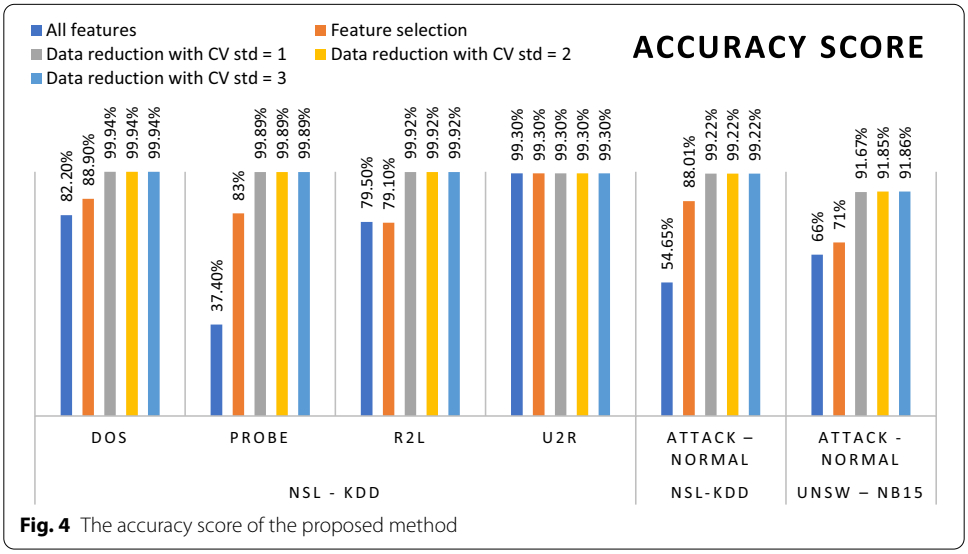

Fig. 4 The accuracy score of the proposed method

raw dataset that applies all features to using the reduced data. In this phase, tenfold cross-validation is performed to evaluate the machine learning model generated from the proposed method. The tenfold cross-validation is a popular method because of its simplicity and less biased characteristics. The DoS class's accuracy score has increased around 6.7–17.74%, with the highest accuracy is of 99.94%. The Probe class's accuracy score has risen around 45.6–62.49%, whose highest accuracy is 99.89%. However, the accuracy score of the R2L class decreases by 0.4% in the feature selection process but periodically goes up in the data reduction pro-

cess by around 20.42%, with the highest score is 99.92%. Furthermore, the accuracy score of the U2R class is stable at 99.3%. It is likely caused by the minimum number of samples in the U2R that caused imbalanced data in this class. So, the sample variance in the U2R class is too low. While in the attack–normal class, the use of the UNSW-NB15 dataset shows that the proposed method can increase the accuracy score from 66 to 91.86% of the highest accuracy score. As for the NSL-KDD dataset, it can increase the accuracy from 54.65 to 99.22%.

ii. Sensitivity

Figure 5 is the sensitivity score from the research experiments. The sensitivity value represents how many data that detected normal activity are the real normal. In the DoS class, the highest sensitivity score is 78% using the selected features. It means that within 100% normal data, 78% is the real normal activity, and 22% is the attack that is detected as normal. In the Probe class, the use of all features can obtain 91% of the highest sensitivity score; while in the R2L class, the score is 10% and 11% for all features and selected features, respectively. As for the U2R class, there is an increase from 10 to 26%, considering that it is still relatively low. Furthermore, in the attack–normal class using the UNSW-NB15 dataset, the proposed method can increase the sensitivity to 80% of the highest score but using the NSL-KDD dataset, it falls to 69.41%. The proposed method shows that the sensitivity value is not optimal for each scenario. It could be due to the uneven distribution of each class, and it could also be due to biased data that cause detection errors in the system.

iii. Specificity

Figure 6 is the specificity value that represents how many data from all detected attacks are real. By using the DoS class, it is shown that the highest specificity score is 97%. It means that 97% of them are the actual attack, and the rest is normal that is misclassified as attacks. The probe class depicts that the highest score is 88%. The probe class shows the specificity score comparatively low compared with the other class because in the probe class, the data variance in the probe class is rela-

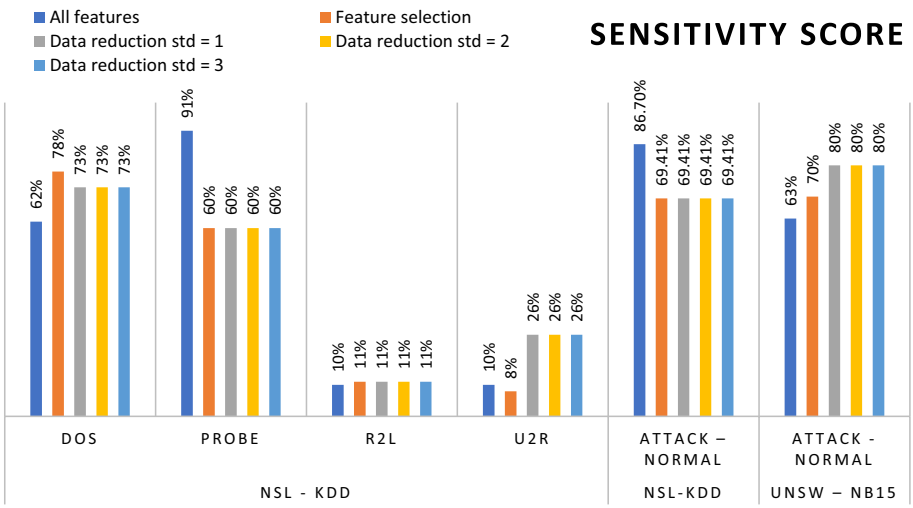

Fig. 5 The sensitivity score of the proposed method

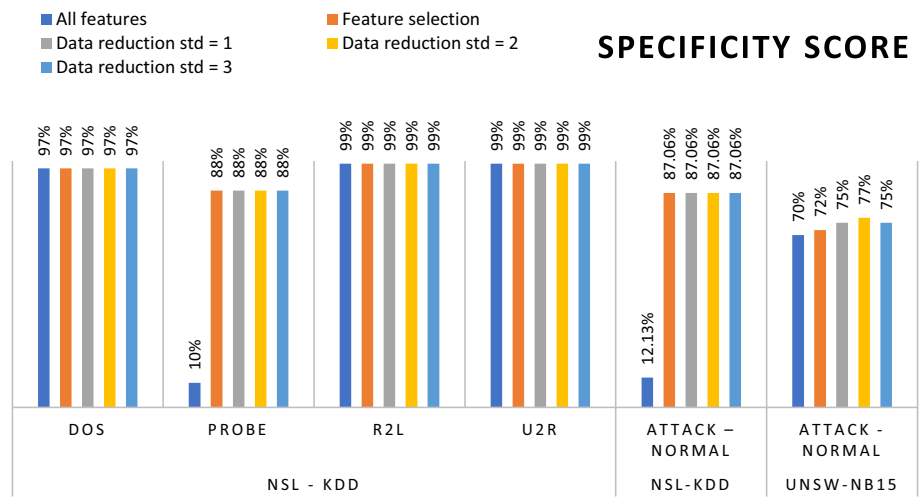

Fig. 6 The specificity score of the proposed method

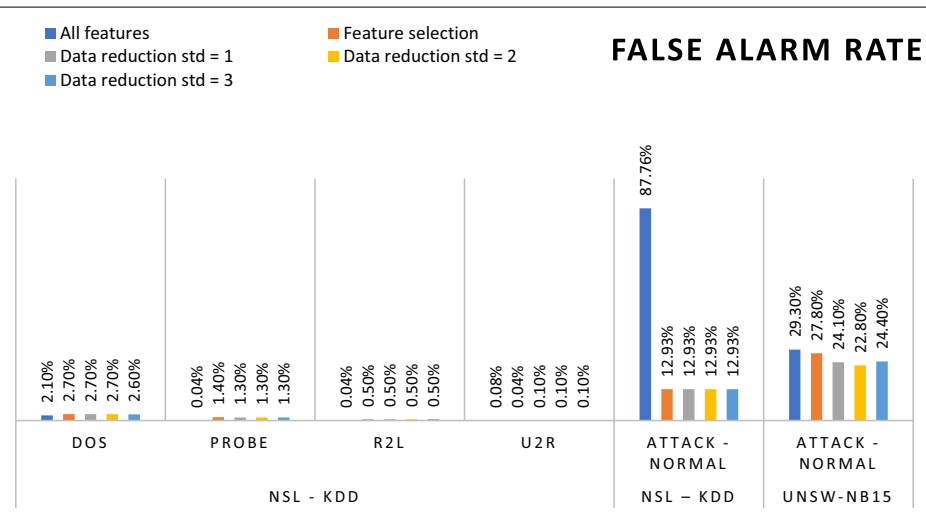

Fig. 7 The false alarm rate score of the proposed method

tively low and each value of each feature in the probe class is relatively close. So, it makes the detection process can be biased, whether it is normal or probe attack. Compared to non-feature selection, the reduced feature significantly raises the score. Regarding the R2L and U2R classes, there is not much difference between using all features and selected features only, where the specificity score is stable at 99%. Furthermore, the proposed method works on both the UNSW-NB15 and NSL-KDD datasets.

iv. False alarm rate

False alarm, as provided in Fig. 7, represents how many attack activities that incorrectly detected. The smaller the value, the better the method. In this experiment, the false alarm rate goes up to around 1–2% for the NSL-KDD dataset and goes

Table 6 The computational times of the proposed method

Dataset	Class	All features (s)	Feature selection (s)	Data reduction (s)		
				std = 1	std = 2	std = 3
NSL-KDD [13]	DoS	1.72	0.231	0.475	0.3	0.32
	Probe	0.89	0.324	0.39	0.39	0.39
	R2L	0.8	0.31	0.37	0.37	0.4
	U2R	0.6	0.19	0.14	0.14 s	0.14
NSL-KDD [13]	Attack–normal	2.47	1.15	1.15	1.15	1.15
UNSW-NB15 [27]	Attack–normal	4.56	1.26	1.45	1.52	1.54

Table 7 Accuracy comparison using the NSL-KDD dataset [13]

Paper	Methods	Accuracy (%)			
		DoS	Probe	R2L	U2R
Zhang et al. [32]	Autoencoder (ANN Deep learning)	83.95	80.38	11.26	32.84
Nkiama et al. [19]	ANOVA F-test + RFE	99.9	99.8	99.88	99.9
Revathi and Malathi [33]	CFS + Random Forest	99.1	98.9	98.7	97.9
Benaddi et al. [34]	PCA-Fuzzy Clustering KNN	94.23	78.86	80.09	69.87
Megantara and Ahmad [28]	Feature importance + RFE	88.98	91.18	81.29	99.42
	Feature importance + RFE + CV 10	99.52	98.9	94.99	99.65
Lian et al. [35]	Ensemble Decision Tree + RFE	99.74	99.2	98.21	99.77
Hussain et al. [31]	Hybrid SVM—ANN	100	99.9	77.4	88.6
Jia et al. [36]	New Deep Neural Network (NDNN)	98.67	97.73	96.94	81.82
Proposed method	Hybrid Machine Learning Method	99.94	99.89	99.89	99.22

down for the UNSW-NB15. The increase of this rate is likely caused by relevant features that are classified as irrelevant. The number of incorrectly detected packets can be caused by biased or outlier data in the dataset; it misclassifies in the IDS detection.

v. Computational time

Computational time represents how long the system processes the data. Indirectly, it also measures the reliability of the system. In this study, the result is given in Table 6. From this table, we find that the proposed method can be implemented in the real system. Moreover, current hardware technology is much advanced than ours. It is also depicted that the proposed method can decrease the computation time from the system by 1–3 s for both NSL-KDD and UNSW-NB15 datasets. This lower number is influenced by the reduced number of features and anomaly/outlier data. The computational time value is hardware-dependent in that the hardware specification influences it. However, with the minimum hardware specification as it is done in this research, the computational time shows exceptional value. According to these experiment values, we believe that the proposed method can be applied in different hardware specifications, whether low or high.

Table 8 Accuracy comparison using the UNSW-NB15 dataset [27]

Paper	Methods	Accuracy (%)
Nawir et al. [37]	Average One Dependence Estimator (AODE)	94.37
	Bayesian Network (BN)	92.7
Kasongo and Sun [38]	Feature Norm. + XGBoost + DT	90.85
	Feature Norm. + XGBoost + ANN	84.39
	Feature Norm. + XGBoost + LR	77.64
	Feature Norm. + XGBoost + KNN	84.46
	Feature Norm. + XGBoost + SVM	60.89
Belouch et al. [39]	RepTree	88.95
Roy and Cheung [40]	BLSTM RNN	95.71
Viet et al. [41]	Deep Belief Network	99.45
Jing and Chen [42]	SVM Modeling	85.99
Proposed method	Hybrid Machine Learning Method	91.86

Method comparisons

Tables 7 and 8 provide a comparison between the proposed method and recent research using the NSL-KDD and UNSW-NB15 datasets, respectively. The data are obtained by performing tenfold cross-validation on the training dataset. As its characteristics, Table 8 compares the performance using only two classes: attack and normal. Table 7 shows the accuracy comparison between the proposed and the previous methods using a similar NSL KDD dataset and similar class categorization. In the DoS class, the proposed method produces the highest accuracy score, 99.94%, and gets a better accuracy than eight previous research from nine compared research. The proposed method produces the highest accuracy score of 99.89% in the Probe class and gets a better accuracy score than 8 from 9 previous research. Both DoS and Probe cannot get through with the research in [31], which gets an accuracy score of 100% in DoS, and 99.9% in probe class. However, their accuracy score in R2L and U2R drops drastically, while our proposed method gets the stable value of 99.89% for R2L and 99.22% for the U2R class. The proposed method provides a stable accuracy score for each class and better accuracy than most evaluated previous research. However, the U2R class accuracy score shows unsatisfactory results due to the imbalanced data in this class.

Table 8 shows the accuracy of the proposed method and the previous research with a similar UNSW-NB15 dataset for a two-class attack–normal class. The proposed method produces an accuracy score of 91.86% and gets a better accuracy score from 7 to 11 previous research methods even it still needs more improvement. The proposed method applied in the NSL-KDD delivers a better accuracy score than that in the UNSW-NB15 dataset because the ratio of total amount data of each attack class in UNSW-NB15 is less proportionally distributed (imbalanced data) than that in the NSL-KDD. So, it makes the data diversity/variance in the dataset relatively low, which can cause possible biased data in it. It is shown that the proposed method is competitive enough and gets better evaluation than the previous research method, although some improvements will make it much better.

Conclusions

In this research, we have proposed some mechanisms to increase the performance of IDS detection. The problem in IDS is the considerable number of data and features that cause the irrelevant features and outlier data in the dataset. To handle that problem, we introduce a hybrid machine learning mechanism, which combines the feature selection process representing supervised learning with the data reduction process as the unsupervised learning method. In order to limit the number of features in the ensemble-based feature selection process, we calculate the median non-zero data from feature importance ranking. Furthermore, for configuring the boundary of the outlier data, we propose a standard normal distribution/Gaussian distribution algorithm on the LOF score for separating the normal from outlier data.

In this research, some evaluations have also been done to measure the capability of the proposed method. Based on the experimental results, we find that reducing the number of features by selecting only the relevant ones can improve the system's performance. Besides, separating important features leads to better results, specifically in the wrapper-based feature selection method. Moreover, the Local Outlier Factor (LOF) method can detect the local outlier data, and the Gaussian distribution is to configure the cut-off value.

Based on the experimental results, it is found that the proposed method can increase the accuracy of the system comparing with both that without implementing the proposed method and the previous research. This method can also reduce the processing time. However, the sensitivity, specificity, and false alarm rate still need more improvement to enhance the performance. The number of biased, imbalanced, and outlier data can be further optimized to increase IDS performance.

In the future, mechanisms to configure the cut-off value for detecting the outlier data should be enhanced, imbalanced data in several classes should be handled, and the LOF cluster size can be optimized. As depicted in the experiment, this value affects the overall performance of the system.

Abbreviations

ARPANET: Advanced Research Projects Agency Network; DoS: Denial-of-Service; ELM: Extreme Learning Machine; HIDS: Hybrid Intrusion Detection System; IDS: Intrusion detection system; LOF: Local Outlier Factor; MitM: Man-in-the-Middle; NBFS: Naïve Base feature selection; R2L: Remote to user; RD: Reachability distance; U2R: User to root.

Acknowledgements

This research is supported by the Ministry of Education, Culture, Research and Technology, the Republic of Indonesia.

Authors' contributions

AAM: design, software, experiment, evaluating, analysis, writing draft; TA: analysis, editing draft, funding acquisition, supervision. All authors read and approved the final manuscript.

Funding

Institut Teknologi Sepuluh Nopember; the Ministry of Education, Culture, Research and Technology, the Republic of Indonesia (1264/PKS/ITS/2021).

Availability of data and materials

On request.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication
Not applicable.

Competing interests
Authors have no competing interests.

Received: 1 July 2021 Accepted: 23 October 2021
Published online: 02 November 2021

References

1. Ray PP. A survey on Internet of Things architectures. J King Saud Univ Comput Inf Sci. 2018;30(3):291–319.
2. Izuakor C. Understanding the impact of cyber security risks on safety. In: ICISSP 2016—Proc 2nd Int. Conf. Inf. Syst. Secur. Priv., no. Icissp. 2016. pp. 509–13.
3. Kumar DA. Intrusion detection systems: a review. Int J Adv Res Comput Sci. 2017;8(8):356–70.
4. Othman SM, Alsohybe NT, Ba-Alwi FM, Zahary AT. Survey on intrusion detection system types. Int J Cyber-Secur Digit Forensics. 2018;7(4):444–62.
5. Jacob NM, Wanjala MY. A review of intrusion detection systems. Glob J Comput Sci Technol. 2017;17(3):11–4.
6. Jyothsna V, Rama Prasad VV, Munivara Prasad K. A review of anomaly based intrusion detection systems. Int J Comput Appl. 2011;28(7):26–35.
7. Sen J, Mehtab S. Machine learning applications in misuse and anomaly detection. In: Security and privacy from a legal, ethical, and technical perspective. pp. 1–15. 2020.
8. L'Heureux A, Grolinger K, Elyamany HF, Capretz MAM. Machine learning with big data: challenges and approaches. IEEE Access. 2017;5(May):7776–97.
9. Jovic A, Brkic K, Bogunovic N. A review of feature selection methods with applications. In: 2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron., vol. 112, no. May, pp. 25–9. 2015.
10. Saini O, Sharma S. A review on dimension reduction techniques in data mining. Comput Eng Intell Syst. 2018;9(1):7–14.
11. Ernst M, Haesbroeck G. Comparison of local outlier detection techniques in spatial multivariate data. Data Min Knowl Discov. 2017;31(2):371–99.
12. Eid HF, Hassanien AE, hoon Kim T, Banerjee S. Linear correlation-based feature selection for network intrusion detection model. In: Commun. comput. inf. sci., vol. 381 ccis, pp. 240–48. 2013.
13. Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set in Computational Intelligence for Security and Defense Applications. In: Comput. Intell. Secur. Def. Appl., no. Cisda, pp. 1–6. 2009.
14. Amiri F, Rezaei Yousefi M, Lucas C, Shakery A, Yazdani N. Mutual information-based feature selection for intrusion detection systems. J Netw Comput Appl. 2011;34(4):1184–99.
15. Mohammed MN, Ahmed MM. Data preparation and reduction technique in intrusion detection systems: ANOVA-PCA. Int J Comput Sci Secur. 2019;13(5):167–82.
16. Almasoudy FH, Al-Yaseen WL, Idrees AK. Differential evolution wrapper feature selection for intrusion detection system. Procedia Comput Sci. 2020;167(2019):1230–9.
17. Zhou Y, Cheng G, Jiang S, Dai M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. Comput Netw. 2020;174:107247.
18. Aljawarneh S, Aldwairi M, Yassein MB. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. J Comput Sci. 2018;25:152–60.
19. Nkiama H, Zainudeen S, Saidu M. A subset feature elimination mechanism for intrusion detection system. Int J Adv Comput Sci Appl. 2016;7(4):148–57.
20. Iman AN, Ahmad T. Data reduction for optimizing feature selection in modeling intrusion detection system. Int J Intell. 2020;13(6):199–207.
21. Prasad M, Tripathi S, Dahal K. Unsupervised feature selection and cluster center initialization based arbitrary shaped clusters for intrusion detection. Comput Secur. 2020;99:102062.
22. Pu G, Wang L, Shen J, Dong F. A hybrid unsupervised clustering-based anomaly detection method. Tsinghua Sci Technol. 2021;26(2):146–53.
23. Saleh AI, Talaat FM, Labib LM. A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. Artif Intell Rev. 2019;51(3):403–43.
24. Gupta N, Vaisla KS, Kumar R. Design of a structured hypercube network chip topology model for energy efficiency in wireless sensor network using machine learning. SN Comput Sci. 2021;2(5):1–13.
25. Gupta N, Jain A, Vaisla KS, Kumar A, Kumar R. Performance analysis of DSDV and OLSR wireless sensor network routing protocols using FPGA hardware and machine learning. Multimed Tools Appl. 2021;80(14):22301–19.
26. Bay SD, Kibler D, Pazzani MJ, Smyth P. The UCI KDD archive of large data sets for data mining research and experimentation. ACM SIGKDD Explor Newsl. 2000;2(2):81–5.
27. Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015—Proc. 2015.
28. Megantara AA, Ahmad T. Feature importance ranking for increasing performance of intrusion detection system. In: 2020 3rd Int. Conf. Comput. Informatics Eng. IC2IE 2020, pp. 37–42. 2020.
29. Ronaghan S. The mathematics of Decision Trees, Random Forest and feature importance in Scikit-learn and Spark. 2018. https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3.
30. Breuniq MM, Kriegel HP, Ng RT, Sander J. LOF: identifying density-based local outliers. SIGMOD Rec (ACM Spec Interes Gr Manag Data). 2000;29(2):93–104.

31. Hussain J, Lalmuanawma S, Chhakchhuak L. A two-stage hybrid classification technique for network intrusion detection system. Int J Comput Intell Syst. 2016;9(5):863–75.
32. Zhang C, Ruan F, Yin L, Chen X, Zhai L, Liu F. A deep learning approach for network intrusion detection based on NSL-KDD dataset. In: Proc. Int. Conf. Anti-Counterfeiting, Secur. Identification, ASID, vol. 2019-Octob, pp. 41–5. 2019.
33. Revathi S, Malathi A. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. Int J Eng Res Technol. 2013;2(12):1848–53.
34. Benaddi H, Ibrahimi K, Benslimane A. Improving the intrusion detection system for NSL-KDD dataset based on PCA-Fuzzy Clustering-KNN. In: Proc.—2018 Int. Conf. Wirel. Networks Mob. Commun. WINCOM 2018, pp. 1–6. 2019.
35. Lian W, Nie G, Jia B, Shi D, Fan Q, Liang Y. An intrusion detection method based on decision tree-recursive feature elimination in ensemble learning. Math Probl Eng. 2020;2020:2835023.
36. Jia Y, Wang M, Wang Y. Network intrusion detection algorithm based on deep neural network. IET Inf Secur. 2019;13(1):48–53.
37. Nawir M, Amir A, Lynn OB, Yaakob N, Badlishah Ahmad R. Performances of machine learning algorithms for binary classification of network anomaly detection system. J Phys Conf Ser. 2018;1018(1):012015.
38. Kasongo SM, Sun Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. J Big Data. 2020;7(1):105.
39. Belouch M, El S, Idhammad M. A Two-stage classifier approach using RepTree Algorithm for network intrusion detection. Int J Adv Comput Sci Appl. 2017;8(6):389–94.
40. Roy B, Cheung H. A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network. In: 2018 28th Int. Telecommun. Networks Appl. Conf. ITNAC 2018, pp. 1–6. 2019.
41. Viet HN, Trang LLT, Nguyen Van Q, Nathan S. Using deep learning model for network scanning detection. In: ACM Int. Conf. Proceeding Ser., no. June 2018, pp. 117–21. 2018.
42. Jing D, Chen HB. SVM based network intrusion detection for the UNSW-NB15 dataset. In: Proc. Int. Conf. ASIC, pp. 1–4. 2019.

Publisher's Note