Applied Intelligence
https://doi.org/10.1007/s10489-018-01408-x

A new hybrid approach for intrusion detection using machine learning methods

Ünal Çavuşoğlu[1]

Abstract
In this study, a hybrid and layered Intrusion Detection System (IDS) is proposed that uses a combination of different machine learning and feature selection techniques to provide high performance intrusion detection in different attack types. In the developed system, firstly data preprocessing is performed on the NSL-KDD dataset, then by using different feature selection algorithms, the size of the dataset is reduced. Two new approaches have been proposed for feature selection operation. The layered architecture is created by determining appropriate machine learning algorithms according to attack type. Performance tests such as accuracy, DR, TP Rate, FP Rate, F-Measure, MCC and time of the proposed system are performed on the NSL-KDD dataset. In order to demonstrate the performance of the proposed system, it is compared with the studies in the literature and performance evaluation is done. It has been shown that the proposed system has high accuracy and a low false positive rates in all attack types.

Keywords Intrusion detection system · Machine learning algorithm · Hybrid system · Feature selection · NSL-KDD

1 Introduction

Along with rapid developments in information and communication technologies, attacks on systems are increasing at a remarkable rate. Preventing attacks on systems and ensuring data security have become one of the most important needs of individuals and institutions. Studies on several different approaches and methods for providing systems and data security are carried out. The provision of data security is made possible by the prevention of data access by people who do not have permission to access the data, the capture of data, and the exchange of data or the prevention of data corruption [1, 2].

An IDS is a system that inspects network traffic data on computer networks to determine harmful activities and alerts when such an activity is detected. Structures used in IDS designed to detect anomalous contents are generally divided into three categories: statistical methods, knowledge-based expert systems, and machine learning techniques. Machine learning techniques are widely used in intrusion detection system designs to detect and prevent attacks. Through this system, sensing the content on normal and abnormal network traffic, it is trying to prevent system damage from the attack. Attack detection systems that study network traffic work with two approaches [3].

Anomaly detection systems try to detect anomalies by examining the system's compliance with normal traffic. Misuse detection systems operate on the signs registered in the system. However, these systems are highly vulnerable to new attacks [4–7]. Misuse detection systems are only effective on known attacks. Successful detections are performed with low error rates in detecting known attacks. Anomaly detection systems continuously monitor the traffic on the network and create a normal traffic pattern, enabling the detection of unknown attacks. If the traffic on the network is far from normal behavior, it is labeled as an attack. In the IDS design where machine learning algorithms are used, the system is trained by using the data obtained in the network traffic and a design is realized for detecting abnormal situations [8].

The literature contribution of this article can be summarized as follows:

- By using feature selection techniques according to protocol type which is one of the most important components of network traffic, it is possible to produce more effective results with less attribute in dataset.

✉ Ünal Çavuşoğlu
unalc@sakarya.edu.tr

[1] Department of Computer Engineering, Sakarya University, Serdivan, Sakarya 54187, Turkey

Published online: 04 February 2019

- In the case of a feature selection technique, proposed is a structure that constructs a new dataset from the combining of selected attributes by using feature selection techniques widely used in the literature.
- Another contribution is to create sub-datasets according to the commonly known attack types on the widely used NSL-KDD dataset to measure the performance of IDS detection systems in the literature. As a result of the tests made with many different algorithms, the most suitable algorithm is determined according to the attack type and a new layered hybrid IDS system in which these algorithms are used together is proposed.
- It has been determined that the system has high success in all attack types with layered architecture which uses the feature selection algorithm according to protocol type and uses the most appropriate algorithm according to the attack type.

The rest of the paper is organized as follows: In Section 2, related works are presented, and machine learning algorithms used in the system, NSL-KDD dataset and feature selection about information are given. In Section 3, the proposed hybrid layered IDS model is presented, data preprocessing and feature selection techniques are explained. In Section 4, evaluation criteria is given and performance tests are conducted and compared with those in the literature. In the last section, the proposed system is evaluated.

2 Background

2.1 Related works

In this section, works on IDS system designs in the literature are presented. Along with the IDS system literature studies, work on future selections techniques has also been included. Table 1 shows the recent hybrid IDS system design studies. Table 1 lists the title of the article, the author and year of publication, the dataset used to test the system, the used machine learning method, the evaluation criteria, and whether or not to use feature selection. When the studies in the literature are examined in Table 1, it is seen that the systems in which many different machine learning techniques are used together have been developed. In the tests of the developed systems, KDD-CUP99, NSL-KDD and DARPA 1998 datasets which are widely used in the literature are used. It has been determined that a number of feature selection techniques have not been used in the given studies.

When looking at the used machine learning techniques, it appears that only a single machine learning algorithm is used instead of a hybrid structure in some studies [9–12].When the literature is examined, it is seen that the IDS designs in which different data mining algorithms are used together [13–24]. Another remarkable characteristic of the proposed system is the realization of designs that use genetic algorithms and data mining algorithms together [25–27]. In addition to the genetic algorithm, it is seen that there are hybrid studies performed with other herustic methods. ANN-GSO [28], ANN-FUZZY [29, 30], ANN [31–34], PSO [35], ANT colony [36, 37], Fuzzy [38]. When the evaluation criteria are examined, it is seen that Detection Rate and Accuracy are preferred among many evaluation criteria in most studies. It is also seen that the training and test times which have a very critical issue for IDS systems, are given as a criterion for evaluation with very little work.

There are also studies in the literature that operate on datasets using different feature selection techniques. Some of these studies are mentioned below. Wang and Feng proposed a hybrid feature selection method using KNN and SVM machine learning algorithms and tested them on different datasets [39]. Mukherjee and Sharma have presented an intrusion detection system design using Correlation-based Feature Selection, Information Gain and Gain Ratio feature selection methods in their work [40]. Amiri et al. have proposed a system in which both linear and non-linear measures and linear correlation coefficients and mutual information are used together for a feature selection process in their work [41].

Manzoor and Kumar proposes an intrusion detection system with ANN Classifier which uses Information Gain and Correlation based feature selection techniques [42]. Madbouly et al. performed feature selection with different methods such as Gain Ratio, Info gain, PSO, Tabu search in the proposed attack detection system and compared the results for different evaluation criteria in all attack types [43]. In Zhang and Wang's work, a new feature selection method based on the Bayesian network is proposed. The performance evaluation of the proposed method and other methods commonly used in the literature are performed on the NSL-KDD dataset [44]. Pervez and Farid have suggested a new system for multi-class intrusion classification tasks by employing the filter method with support vector machine (SVM) classifier on NSL-KDD dataset [45]. Ambusaidi et al. proposed a new filter based feature selection algorithm named by the Modified Mutual Information Based Feature Selection (MMIFS) in their work. Tests of the proposed system on different sets of data have been conducted and compared with other methods in the literature [46]. Kang and Him used the optimization approach for future selection in their work. They propose a method using the KNN algorithm to obtain the optimal subset [47].

Table 1 The IDS works in the literature

Article title	Authors & year	Dataset	Technique	FS	E. Criteria
A new evolutionary neural networks based on intrusion detection systems using multiverse optimization	Benmessahel et al. [34]	NSL-KDD	ANN	No	ACC, DR, FNR
A Hybrid Approach based on Classification and Clustering for Intrusion Detection System	Jasmeen K. Chahal et al. [13]	NSL-KDD	KM, SVM	No	ACC, FPR, FNR
A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset	M. R. Parsaei et al. [20]	NSL-KDD	SMOTE, CANN	Yes	ACC, FNR
A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers	Ahmed I. Saleh et al. [14]	KDD'Cup99, NSL-KDD	NB, SVM, KNN	Yes	DR, Time, RMS
A hybrid method consisting of GA and SVM for intrusion detection system	Shahri B.M. et al. [25]	KDD'Cup99	GA, SVM	Yes	TPR, FPR, Precision, ROC
A hybrid multi-layer intrusion detection system in cloud	Manicka M.(2018) [28]	NSL-KDD	ANN, GSO, TS	No	DR
A hybrid network intrusion detection framework based on random forests and weighted k-means	Reda M. Elbasiony [15]	KDD'Cup99	RF, KM	No	DR, FPR
A multi-level intrusion detection method for abnormal network behaviors	Soo-Yeon Ji et al. [16]	NSL-KDD	SVM, NB, ANN	Yes	ACC
A study on supervised machine learning algorithm to improvise intrusion detection systems for mobile ad hoc networks	S. Vimala [29]	KDD'Cup99	ANN, FL, SVM	No	DR, FPR
Adaboost Ensemble with Genetic Algorithm Post Optimization for Intrusion Detection	Hany M. Harb et al. [26]	NSL-KDD	AB,GA	Yes	ACC, Time
A two-level hybrid approach for intrusion det.	Chun G. [11]	KDD'Cup99	KNN	No	DR, FPR, ACC
An effective combining classifier approach using tree algorithms for network intrusion det.	Jasmin Kevric et al. [21]	NSL-KDD	RT, NBT	No	ACC
An Efficient Hybrid Anomaly Detection Scheme Using K-Means Clustering for WSN	Mohammad Wazid et al. [9]	–	KM	No	DR, FPR
An enhanced J48 classification algorithm for the anomaly intrusion detection systems	Shadi Aljawarneh et al. [10]	NSL-KDD	J48	Yes	ACC

Table 1 (continued)

Article title	Authors & year	Dataset	Technique	FS	E. Criteria
An Intrusion Detection Framework Based on Hybrid Multi-Level Data Mining	Haipeng Yao et al. [22]	KDD'Cup99	SVM,ELM,KM	Yes	ACC, Precision, Recall
An intrusion detection system using network traffic profiling and online sequential extreme learning machine	Raman Singh et al. [12]	NSL-KDD, Kyoto	OS-ELM	Yes	ACC, Precision, Recall, F-Value,Time
An Efficient Hybrid Multilevel Intrusion Detection System in Cloud Environment	Partha Ghosh et al.[31]	NSL-KDD	KNN, ANN	Yes	DR
Anomaly network traffic detection algorithm based on information entropy measurement under the cloud computing environment	Chen Yang [35]	KDD'Cup99, NSL-KDD	SVM, PSO	No	Precision, Recall, F-Value
Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation	V. Balamurugan [33]	—	KM, ANN	No	DR, FPR,
Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system	Wathiq Laftah Al-Yaseen et al. [24]	KDD'Cup99	SVM, KM	No	ACC,DR,FPR
Hybrid decision tree and naive Bayes classifiers for multi-class classification tasks	Dewan Md. Farid et al. [23]	UCI repository 10 datasets	DT, NB, J48	No	ACC, Precision, Sensitivity, Specificity
Mining network data for intrusion detection through combining SVMs with ant colony net.	W. Feng et al. [36]	KDD'Cup99	SVM, ANT	No	DR, FPR, FNR
A novel hybrid intrusion detection method integrating anomaly detection with misuse det.	G. Kim et al. [17]	KDD'Cup99	DT,SVM	No	DR, ROC
Practical real-time intrusion detection using machine learning approaches	P. Sangkatsanee et al. [32]	KDD'Cup99	DT, ANN	Yes	DR
A new approach to intrusion detection using artificial neural networks and fuzzy clustering	G. Wang et al. [38]	KDD'Cup99	FL, ANN	No	Precision, Recall, F-Value

Table 1 (continued)

Article title	Authors & year	Dataset	Technique	FS	E. Criteria
Feature Selection Based Hybrid Anomaly Intrusion Detection System Using K-Means and RBF Kernel Function	Ravale et al. [18]	KDD'Cup99	KM, RBF	No	DR,ACC
Hybrid Modified -Means with C4.5 for Intrusion Detection Systems in Multiagent Systems	Laftah Al-Yaseen et al. [19]	KDD'Cup99	KM, J48	No	ACC,Time
A novel hybrid KPCA and SVM with GA model for intrusion detection	Fangjun Kuang et al. [27]	KDD'Cup99	SVM,	Yes	DR,FPR,ACC
Fuzziness based semi-supervised learning approach for intrusion detection system	Rana A. R. A. et al. [30]	NSL-KDD	FL, ANN	No	ACC,Time
An efficient intrusion detection system based on support vector machines and gradually feature removal method	Yinhui Li et al. (2012) [37]	KDD'Cup99	ANT, SVM	Yes	ACC, MCC, Time

Algorithms → **ANN**: Artifical Neural Network; **KM**:K-means; **SVM**: Support Vector Machine; **CANN**: Cluster Artifical Neural Network; **NB**: Naive Bayes; **GA**: Genetic Algorithm; **TS**: Tabu Search; **RF**: Random Forest; **FL**: Fuzzy Logic; **AB**: AdaBoost; **RT**: Random Tree; **PSO**: Particle Swarm Opt.; **DT**: Decision Tree; **ANT**: Ant Colony Opt.; **SOM**: Self Organizing Map; **OS-ELM**: Online Sequential Extreme Learning Machine; **GSO**: Glow swarm optimization **Evaluation Criteria** → **ACC**: Accuracy ,**DR**: Detection Rate; **FNR**: False Negative Rate; **TPR**: True Positive Rate; **FPR**: False Positive Rate; **MCC**: Matthews Correlation Coefficients

Beulah and Punithavathani have proposed a hybrid approach that combines the best features of different feature selection methods in their work. Six qualities are determined for the NSL-KDD dataset and their performances are compared using different classification approaches [48]. Bhattacharya and Selvakumar is performed clustering using the multi-weight feature selection approach and filter and wrapper future selection methods. Two new weight estimation algorithms are presented in the study [49]. Bajaj and Arora used Information Gain, Gain ratio and correlation attribute evaluation as feature selection method in their study. They have performed tests of different machine learning algorithms with datasets obtained using these techniques [50]. Osaniye et al. have proposed a system for detecting DDoS attacks on the cloud in their work. They have used a method called ensemble-based multi-filter feature selection (EMFFS) in which the feature selection methods are used together [51]. Sethuramalingam and Naganathan have proposed a hybrid future selection algorithm that combines the Information Gain and the genetic algorithm and tested it on the NSL-KDD dataset [52]. Apart from these studies, there are feature selection studies carried out using different approaches and methods in other literature [53–56].

In the literature, many different machine learning techniques are used to improve attack detection performance. Table 1 presents the recent literature on IDS system design studies. When studies in the literature are evaluated, the use of only some of the machine learning techniques in intrusion detection systems is not sufficient for a harmful traffic to be detected with high accuracy and false positive rate in all attack types. In addition, systems designed by combining many different techniques such as Genetic Algorithm (GA), Fuzzy, ANT Colony, Particle Swarm Optimization (PSO) have a very complicated structure and cause training and test times to increase. It has been found that unnecessary and large-size data are used on large datasets since the feature selection techniques are not used a lot of in the literature. As a result of this situation, the processing times are prolonged and the desired performance can not be achieved.

In order to prevent all these disadvantages in this study, a layered and hybrid IDS system using different machine learning algorithms according to different attack types with two different feature selection methods is proposed. Algorithms with high accuracy rate and false positive rates are determined by using machine learning techniques according to attack types and they are used in the system design. In addition, the dataset size has been reduced by two different feature selection techniques, which are a new feature selection method based on the protocol type and a combination of the features selected by different

feature selection algorithms. Unlike the studies in the literature, new feature selection techniques are proposed and higher performance is achieved with less attribute. For each attack type, a different machine learning method has been determined and used in the system design and high performance results have been obtained. Thanks to the proposed system, a high level of accuracy and short-term intrusion detection has been achieved.

2.2 NSL-KDD dataset

The KDD-CUP99 dataset [57] is a widely used dataset for testing systems developed to detect computer network traffic anomalies. This dataset contains many records in different attack types. However, in the studies performed on this dataset, it has been determined that there are some cases that adversely affect the performance of the systems tested in the dataset. To solve this problem, it has been suggested to use a new dataset called NSL-KDD [58] to test the proposed systems, eliminating some records in the KDD CUP99 dataset. In the NSL-KDD dataset, the unnecessary samples from the dataset to be used for training are cleared and the size of the dataset is set to a reasonable value for the anomaly detection. Table 2 shows sample numbers according to the attack types in the NSL-KDD train+ and 20% NSL-KDD train+ sets. The atributes of NSL-KDD 20% training + dataset such as attribute name, data types, description, min and max value values are presented in Table 3. The 20% NSL-KDD train+ dataset is used for the operations performed in this study.

There are 4 different attack types in the NSL-KDD dataset. The description of these types of attacks and attack types found in these attack groups are given below.

- **Denial of Service Attack (DoS):** the result of this attack, the system's resources or network traffic usage is increased to make the system unable to provide service. (*Back*, *Land*, *Neptune*, *Pod*, *Smurf*, *Teardrop*, *Apache2*, *Udpstorm*, *Processtable*, *Worm*)
- **User to Root Attack (U2R):** This account gets access to the root account after an ordinary user account is obtained. ($Buffer\_overflow$, *Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps)*
- **Remote to Local Attack (R2L):** A packet is sent to a machine in the network and a weakness is detected in the network to obtain a user account. ($Guess\_Password$, Ftp_write, *Imap, Phf, Multihop, Warezmaster, Warez client, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named)*
- **Probe Attack:** gathering information about network weaknesses by scanning the network for misuse. ($Satan$, $Ipsweep$, $Nmap$, $Portsweep$, $Mscan$, $Saint$)

2.3 Machine learning technique that use in the works

Many different data mining algorithms have been developed to perform data mining operations on datasets. Some of these are Bayes [59], Naive Bayes [60], Random Forest [61], Fuzzy Logic [62], Decision Trees [63], Artificial Neural Networks [64] and Decision Supporting Machines [65], K Nearest Neighbour [66], K-means [67]. Data mining algorithms perform operations on datasets using very different approaches. The performance of these algorithms differs their performance according to the structure of the datasets. For high performance, an appropriate algorithm selection should be made for the structure of the dataset [68, 69]. In this section, data mining algorithms used for classification operations on the NSL-KDD dataset are briefly described.

2.3.1 Naive Bayes algorithm

The Naive Bayes classification algorithm [60] is a simple algorithm that tries to determine which classes are included by using probability theorems. It is easy to use and can generate predictions with a single scan. The algorithm uses a simplified version of the Bayesian theorem. The conditional probability theory is used in the calculation of the class that will contain a sample in the dataset. The system is trained on the training dataset and the class of the samples in the test dataset is estimated. Although the Naive Bayes algorithm has a simple structure, it produces quite successful results. The probabilistic expressions used in Bayes' theorem are given below.

$P(x|y) \rightarrow$ When y event occurs, probability of event x
$P(y|x) \rightarrow$ When x event occurs, probability of event y
$P(x) and P(y) \rightarrow$ the probabilities of x and y events.

$$P(c|x_1, x_2, ..., x_n) = \frac{P(x_1,x_2,.....,x_n|c)P(c)}{P(x_1,x_2,.....,x_n} \\ P(c|X) = P(x_1|c)P(x_2|c).....P(x_n|c)P(c) \quad (1)$$

Table 2 NSL-KDD the number of records according to attack types

NSL-KDD Dataset	DoS	Probe	R2L	U2R	Normal	Total
NSL-KDD Train+	45927	11656	995	52	67343	125973
%20 NSL-KDD Train+	9234	2289	209	11	13449	25192

Table 3 The atributes of NSL-KDD %20 training+ dataset

No	Attribute name	Data types	Attribute description	Min.	Max.
1	duration	Numeric	Length of the connection	0	42862
2	protocol_type	Nominal	Connection protocol	–	–
3	service	Nominal	Destination service	–	–
4	flag	Nominal	Status flag of the connection	–	–
5	src_bytes	Numeric	Bytes sent from source to destination	0	381709090
6	dst_bytes	Numeric	Bytes sent from destination to source	0	5151385
7	land	Nominal	1 if is from/to the same host/port; 0 otherwise	0	1
8	wrong_fragment	Numeric	Number of wrong fragment	0	3
9	urgent	Numeric	Number of urgent packets	0	1
10	hot	Numeric	Number of hot indicators	0	77
11	num_failed_logins	Numeric	Number of failed login in attempts	0	4
12	logged_in	Nominal	1 if successfully logged in; 0 otherwise	0	1
13	num_compromised	Numeric	Number of compromised conditions	0	884
14	root_shell	Numeric	1 if root shell is obtained; 0 otherwise	0	1
15	su_attempted	Numeric	1 if “su root” command attempted; 0 otherwise	0	2
16	num_root	Numeric	Number of root accesses	0	975
17	num_file_creations	Numeric	Number of file creation operations	0	40
18	num_shells	Numeric	Number of shell prompts	0	1
19	num_access_files	Numeric	Number of operations on access control files	0	8
20	num_outbound_cmds	Numeric	Number of outbound commands in an ftp session	0	0
21	is_host_login	Nominal	1 if the login belongs to the hot list; 0 otherwise	0	1
22	is_guest_login	Nominal	1 if the login is a guest login; 0 otherwise	0	1
23	count	Numeric	Number of conn. to the same host as the current conn. in the past two sec.	1	511
24	srv_count	Numeric	Number of conn. to the same service as the current conn. in the past two sec.	1	511
25	serror_rate	Numeric	% of conn. that have “SYN” errors (same-host conn.)	0	1
26	srv_serror_rate	Numeric	% of conn. that have “SYN” errors (same-service conn.)	0	1
27	rerror_rate	Numeric	% of conn. that have “REJ” errors (same-host conn.)	0	1
28	srv_rerror_rate	Numeric	% of conn. that have “REJ” errors (same-service conn.)	0	1
29	same_srv_rate	Numeric	% of conn. to the same service (same service conn.)	0	1
30	diff_srv_rate	Numeric	% of conn. to different services	0	1
31	srv_diff_host_rate	Numeric	% of conn. to different hosts (same-service conn.)	0	1
32	dst_host_count	Numeric	% Count of conn. having the same destination host	0	255
33	dst_host_srv_count	Numeric	% Count of conn. having the same destination host and using the same service	0	255
34	dst_host_same_srv_rate	Numeric	% of conn. having the same destination host and using the same service	0	1
35	dst_host_diff_srv_rate	Numeric	% of different services on the current host	0	1
36	dst_host_same_src_port_rate	Numeric	% of conn. to the current host having the same port	0	1
37	dst_host_srv_diff_host_rate	Numeric	% of conn. to the same service coming from different hosts	0	1
38	dst_host_serror_rate	Numeric	% of conn. to the current host that have an SO error	0	1
39	dst_host_srv_serror_rate	Numeric	% of conn. to the current host and specified service that have an SO error	0	1
40	dst_host_rerror_rate	Numeric	% of conn. to the current host that have an RST error	0	1
41	dst_host_srv_rerror_rate	Numeric	% of conn. to the current host and specified service that have an RST error	0	1

In (1), the classification model of the Naive Bayes algorithm is given. In the equation, c is the specified target and x is the all attributes. When the formula is examined, it is seen that the calculation is performed by performing the multiplication of all conditional probabilities in the Naive Bayes method. After the calculation is done for all classes in the Naive Bayes classifier, the value with max probability is determined and the instance is added to that class.

2.3.2 Random Forest algorithm

The random forest algorithm (RF) [61] is an algorithm that is developed by combining the results of a large number of decision trees trained with different training clusters. It was developed by Breiman in 2001 as an algorithm that uses multiple classification techniques. In the random forest algorithm, different sub-training clusters are created. Preloading is performed in the creation of training clusters. For the expansion of the trees, a method in which the properties are selected at random is used. In the algorithm's operation, each node is divided into branches using the best value among the randomly selected values from each node. Derived trees are obtained by randomly selected variables. The Classification And Regression Trees (CART) algorithm is used for the tree development process from the obtained datasets. The sample to be classified is tagged according to each generated tree and the assigned classes are collected. The instance to be processed is included in the class to which it is assigned the most. Although pruning is found in the CART algorithm, pruning is not performed in the RF algorithm. The lack of pruning in the RF algorithm contributes to the RF algorithm being more successful than the other decision tree methods. Despite the use of multiple tree structures in the RF algorithm, the algorithm is quite fast, it can work with many tree structures, and its performance is better than other decision tree methods [70].

In the algorithm, 2/3 of the training dataset is used for preloading and this dataset is used in tree development. The remaining 1/3 of the training dataset is called out-of-bag data and this dataset is used to test for errors. The branch to be generated from each node is determined by the GINI idex value used in the CART algorithm. In the tree development process, the number of variables to be used in the node and the number of trees to be produced are used as parameters. Figure 1 shows the block diagram of the general operating principle of the RF algorithm.

2.3.3 J-48 (C4.5) Decision Tree algorithm

This algorithm is known in the literature as C 4.5 [63]. In Weka data mining, this algorithm is called J-48. It has emerged with the development of an algorithm known as ID3. The algorithm uses the divide and conquer approach. Unlike the ID3 algorithm, normalization operations are included in this algorithm. The information gain values are calculated in the algorithm and these values are used as a ratio. It is possible to construct lower trees at the creation of the decision tree and to move these lower trees at different levels. In the decision tree algorithm, branch pruning is also performed to delete the problematic data in the dataset from the tree and to reduce the error rate. In the tree building process, a single node is detected and processing is started, and if all of the samples are in one class, the corresponding node is detected as leaf and represents a class. If the node has properties belonging to different classes, the attribute to determine the best segmentation is determined and branching continues.

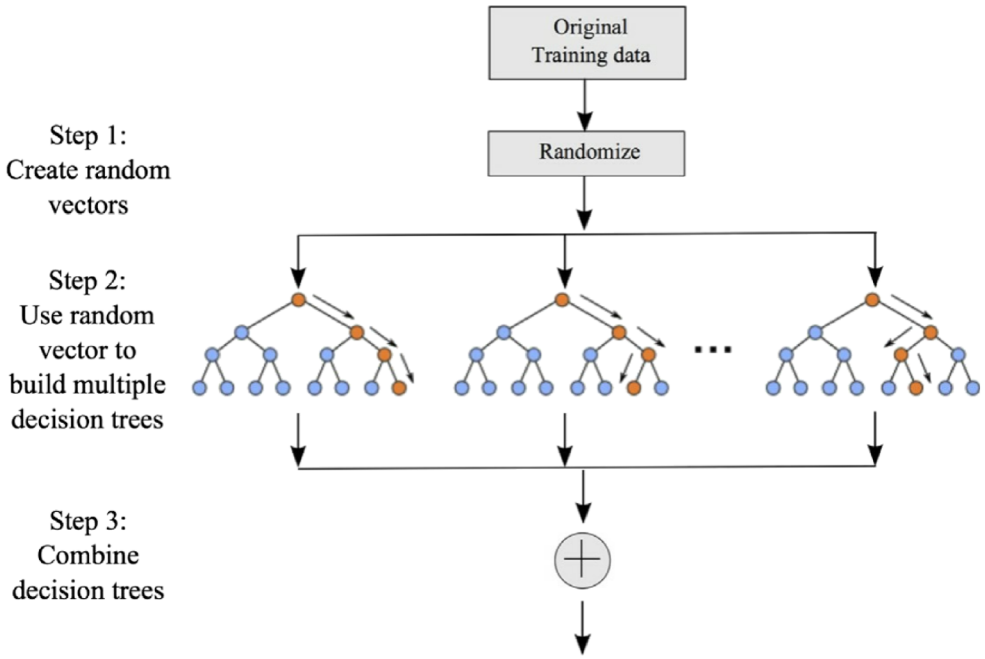

Fig. 1 The random forest algorithm block diagram [71]

The steps of the algorithm are briefly summarized below:

- The information gain values of all features are calculated,
- The attribute with the best information gain value is determined as the decision node in the tree,
- The process continues with the creation of a new sub tree under the determined decision node.
- If the same values are obtained for all elements in the subgroups specified here, the process stops and the final value is determined as the output value. If there is only one node in the subgroup and no distinguishable attribute is found, the process is stopped.

2.3.4 k-Nearest Neighbor (KNN) algorithm

The KNN (k-Nearest Neighbor) classification algorithm [66] is one of the most widely used of machine learning algorithms. In large datasets, processing times may be longer than other algorithms, but produce successful results. In the algorithm, the distance to each sample in the training set is calculated for the data to be tested. Where k is the number of the nearest neighbors to be considered in determining the class. K is determined as an odd number such as 1, 3, 5, and the new class of the sample is determined according to the class of the calculated nearest neighbors. Selecting a value of K higher or lower affects the results in some cases. It is very important to determine the appropriate k value according to the studied dataset. There are different formulas used in neighboring nodes and distance calculation: Euclidean, Manhattan and Minkowski. These formulas are given below.

$$\begin{aligned} Euclidean &\rightarrow \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2)} \\ Manhattan &\rightarrow \sum_{i=1}^{k}|x_i - y_i| \\ Minkowski &\rightarrow \left[\sum_{i=1}^{k}(|x_i - y_i|^q)\right]^{1/q} \end{aligned} \tag{2}$$

2.3.5 Stacking-ensemble technique

Ensemble technique is a method that is used to perform more efficient and efficient classification by using more than one machine learning algorithm together. In this method, a series of learning processes are realized with different machine learning techniques and the results obtained are combined and classified. There are two general operations in the algorithm used. Firstly, distribution of a simple model is produced on the subsets obtained from the original dataset. This is followed by combining the distribution in one aggregated model and obtaining the results. Unlike ordinary machine learning approaches, there is a model production in the stacking method. The created models using the training dataset are combined [72, 74].

The algorithm's work can be summarized as follows:

- In the education phase, the production of the models is carried out by using the dataset and algorithm to be used for training.
- For each obtained model, the samples in the training dataset are labeled.
- The final model, which is the composition of the other models, is obtained from the training dataset using the combiner method.
- After obtaining the final model, the samples in the test dataset are classified using the specified models.
- After classifying each sample in the test dataset, the final prediction is made using the final model and the class predicted by the stacking algorithm of the sample is determined.

There are 3 different methods known as ensemble technique in the literature. Bagging, Boosting and Stacking [75]. These techniques differ from each other in the data mining methods, and the combiner model building functions. While the bagging algorithm tries to minimize the variance value, boosting increase predictive force and stacking aims to achieve both of them. In the function used to obtain a single model, average weight in bagging method, weighted majority vote in boosting method and Logistic regression in stacking method are used.

2.4 Feature selection

Feature selection is one of the most important steps in data mining applications. The determination of the proper feature selection algorithm and its use in operations has an effect that will increase the performance of the application. It also has the effect of decreasing the operational load as it reduces the number of attributes on the dataset and establishes new connections between attributes [76–78]. However, there is no single method for feature selection. The method to be used may vary according to the state of the dataset. The main problem in feature selection is to select the feature that can best distinguish between classes. Different feature selection algorithms may be more appropriate for different sets of data. There are many different algorithms used in the feature selection approach [79–84]. Two methods commonly used in the literature in feature selection processes are briefly described here.

The CfsSubsetEval algorithm [85, 86] performs a selection among the attributes in the dataset that those are highly

related to the class and that are less important. In this way, the most important features of the dataset are identified. CfsSubsetEval method uses a search algorithm. Among the attributes in the algorithm, it is possible to identify those who have the best relationship with the class label. In this case, the specified attribute group has a high correlation with the class tag, but it is determined that the other properties have a less significant status. Therefore, it is expected that the use of attribute group with high linkage with class label in data mining processes will increase the success. In this article, BestFirst search algorithm is used in CfsSubSetEval algorithm.

In the wrapper method [87], subset selection is performed using machine learning algorithms. The machine learning algorithm to be used for attribute selection must be determined in this method. Feature selection algorithms based on filtering methods are evaluated according to statistical tests, whereas machine learning algorithms are used in this method. In this method, the use of different algorithms is used to create subsets with different attributes and the achievements of these subsets are evaluated. The method itself has a predictive approach. Although it varies according to the used algorithms, the Wrapper method generally requires longer processing time than the filtering method but it produces better subsets. As a first step in the selection features, different search techniques such as best search and random search are used to obtain better subsets. In this process all possible subsets are obtained in the whole vector space. Then, the subset which may have optimum performance from the possible subsets, is obtained by the predictor.

3 Proposed hybrid-layered IDS model

The proposed intrusion detection system has been introduced in detail in this section. Firstly, data preprocessing is performed on the NSL-KDD dataset which will be used for system training and testing. After these operations, feature selection procedures are performed using different feature selection techniques and new approaches. Finally, the proposed layered hybrid structure is introduced as a whole.

3.1 Data preprocessing

Transformation and normalization operations have been performed on the NSL-KDD dataset to ensure that the dataset is cleaned from unnecessary data and produce higher performance results. In this case, a more optimal dataset is obtained.

Transformation operation The nominal values in the NSL-KDD dataset have been converted to numeric values. The numeric values of protocol_type, service, and flag attributes in the dataset have been converted. In this way, a dataset consisting entirely of numeric values is obtained and these values are processed as numeric values during classification operations. Table 4 shows nominal values and numeric equivalents used in the conversion process. Protocol types tcp, udp, and icmp are converted to 1, 2, and 3, respectively, and service and flag attributes are respectively converted to numeric values.

The description of the qualities that are transformed is given below.

- protocol_type: Describes the protocol information used for the connection.
- service: Describes network service information used during connection
- flag: Describes the information of connection status

Tables 5 and 6 provide examples showing the cases before and after the transformation process.

Normalization operation Dataset normalization is a very important preprocessing technique, especially in classification. Normalization is used to transform the attributes of the dataset into values compatible with one another. Normalization helps to speed up the operations on the dataset and produce successful results at a higher rate. In the study, min-max normalization is performed on the dataset for normalization. In this method, the largest and smallest values in a group are used. All other data are normalized to these values. The purpose here is to normalize the smallest value to 0 and the largest value to be 1 and to spread all the other

Table 4 Transformation operation values

Attribute name	Nominal value(old)	Numeric value(new)
Protocol type	tcp	1
	udp	2
	icmp	3
Service	aol,..,.,. . .,Z39_50	1-66
Flag	OTH	1
	REJ	2
	RSTO	3
	RSTOS0	4
	RSTR	5
	S0	6
	S1	7
	S2	8
	S3	9
	SF	10
	SH	11

Table 5 Before transformation

No	Instances
1	0,tcp,ftp_data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,0.00,normal
2	0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1.00,1.00,0.00,0.00,0.05,0.07,0.00,255,26,0.10,0.05,0.00,0.00,1.00,1.00,0.00,0.00,DoS
3	0,tcp,ftp_data,SF,334,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,2,20,1.00,0.00,1.00,0.20,0.00,0.00,0.00,0.00,R2L
4	0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,1,16,1.00,0.00,1.00,1.00,0.00,0.00,0.00,0.00,Probe
5	98,tcp,telnet,SF,621,8356,0,0,1,1,0,1,5,1,0,14,1,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,4,0.02,0.02,0.00,0.00,0.00,0.00,0.00,0.00,U2R

Table 6 After transformation

No	Instances
1	0,1,18,10,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,0.00,normal
2	0,1,48,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1.00,1.00,0.00,0.00,0.05,0.07,0.00,255,26,0.10,0.05,0.00,0.00,1.00,1.00,0.00,0.00,DoS
3	0,1,18,10,334,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,2,20,1.00,0.00,1.00,0.20,0.00,0.00,0.00,0.00,R2L
4	0,3,12,10,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,1,16,1.00,0.00,1.00,1.00,0.00,0.00,0.00,0.00,Probe
5	98,1,59,10,621,8356,0,0,1,1,0,1,5,1,0,14,1,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,4,0.02,0.02,0.00,0.00,0.00,0.00,0.00,0.00,U2R

Table 7 After normalization

No	Instances
1	0,0,0.235294,0.9,0.000001,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.001961,0,0,0,0,1,0,0,0.588235,0.098039,0.17,0.03,0.17,0,0,0,0.05,0,normal
2	0,0,0.676471,0.5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.239216,0.009804,1,1,0,0,0.05,0.07,0,1,0.101961,0.1,0.05,0,0,1,1,0,0,DoS
3	0,0,0.235294,0.9,0.000001,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0.001961,0,0,0,0,1,0,0,0.007843,0.078431,1,0,1,0.2,0,0,0,0,R2L
4	0,1,0.147059,0.9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0.003922,0.062745,1,0,1,1,0,0,0,Probe
5	0.002286,0,0.838235,0.9,0,0.000002,0.001622,0,0,1,0.012987,0,1,0.005656,1,0,0.014359,0.025,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0.015686,0.02,0.02,0,0,0,0,0,U2R

data to this 0-1 range. The dataset samples after normalization are given in Table 7. The formula used to calculate the new value is given in (3).

3.2 Proposed feature selection methods

Feature selection algorithms enable the elimination of unnecessary data in datasets, reduce their size and make them more efficient. In this section, feature selection is performed on the NSL-KDD dataset. Two different feature selection methods are presented and two datasets are created using these methods. These datasets are used for training and testing of the proposed system. For the classification on the dataset, the technique of combining attributes with different feature selection techniques has been used to select attributtes with high deterministic properties, to include them in the generated new dataset and to avoid neglecting them. Among the feature selection techniques that are widely used in the literature and obtained high performance as a result of the tests made are included in the proposed system in obtaining new datasets.

3.2.1 The combining method of different feature selection algorithm

In this method, the feature selection process is performed by using the CfsSubsetEval and WrapperSubsetEval feature selection algortihms which are widely used in the literature and described in Section 2. The block diagram of the proposed method is shown in Fig. 2. The CfsSubsetEval algorithm is used for future selection operation on the NSL-KDD 20% training dataset with 41 attributes using BestFirst search technique. WrapperSubsetEval algorithm is used to perform feature selection with different classifiers. Table 8 shows the attributes determined by these algorithms. Following the selection of the different feature selection algorithms, a new dataset with 25 attributes is obtained by combining the selected attributes. In this method, it is aimed to obtain a dataset for an effective classification operation by combining the attributes determined by different algorithms. As a result of the performed operations, the number of attributes of the NSL-KDD dataset with 41 attributes has been reduced to 25.

$$X_{new} = \frac{(X - X_{min})}{X_{max} - X_{min}} \tag{3}$$

3.2.2 The combining method of different feature selection algorithm according to protocol type

In this method, feature selection process is performed according to the protocol type information which is a very important feature in determining the contents of data

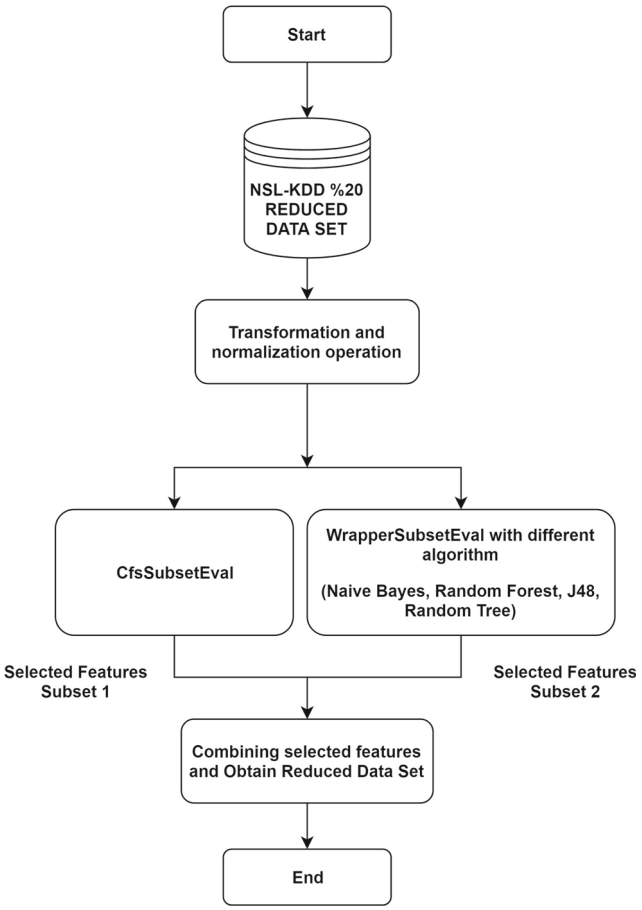

Fig. 2 The block diagram of combining attribute selected different algorithm

traffic in computer networks. Unlike the other feature selection method, this method uses sub datasets that are created according to protocol type. The reason for choosing according to the protocol type is that the protocol information is one of the most important components in terms of traffic within the attributes. In this respect, it is aimed to select attributes with high priority of traffic. The block diagram of the proposed method is shown in Fig. 3. Before the feature selection operation, the NSL-KDD 20% training dataset is divided into sub-datasets according to the protocol types. When examining the sample numbers according to the NSL-KDD 20% dataset protocol types, it is seen that there are 20526 tcp, 3011 udp and 1655 icmp samples. CfsSubsetEval and WrapperSubsetEval feature selection methods which are widely used in the literature are used on the datasets formed according to the protocol type. Table 9 shows the used feature selection methods and the determined attributes using these methods. After the selection process according to each protocol, all the selected features are combined to obtain a dataset with the new reduced number of attributes. With this method, the NSL-KDD 20% training dataset with 41 attributes is reduced to 20 features and the number of attributes is reduced by approximately half. With the use of this dataset, it is expected that the training and testing processes will be performed much faster and with high performance rates.

3.3 The proposed hybrid layered IDS system

The proposed IDS in this section has been introduced in detail. As described in the previous section, after the data preprocessing operations are performed, two different reduced datasets are obtained by applying the proposed new feature selection methods. These datasets obtained are used for the training and testing of the proposed system. Different machine learning algorithms are tested according to each attack type, and the algorithms with the highest accuracy, detection rate performance and lowest error rate are determined according to attack types in the proposed system. The selected algorithms according to attack type are used in system design. The block diagram of the proposed system is shown in Figs. 4 and 5. The datasets obtained after the data preprocessing and future selection operations are divided into sub datasets according to the attack type for training and testing operations. By combining attack and normal traffic types, 4 sub datasets are obtained. Table 10 shows the sample numbers of the new datasets.

The cross fold validation method [88] is used for performance evaluation of the system. In this technique, the dataset is randomly divided into determined k parts. In each iteration a part is used for test operation and the remaining k-1 piece for training. In this way, each part is used for testing and the entire dataset is tested. In this study, all

Table 8 Different feature selection algorithm results

Future selection method	Classifier	Selected futures for each method
CfsSubsetEval	–	3,4,6,8,12,14,25,29,30,37,39
WrapperSubsetEval	Naive Bayes	2,8,12,29,35
WrapperSubsetEval	Random Forest	3,5,6,35,36,38,40
WrapperSubsetEval	J-48	3,4,5,6,8,10,12,16,23,24,25,27, 30,32,33,34,35,38
WrapperSubsetEval	Random Tree	2,3,4,5,6,8,23,30,32,36,38,39,40
Overall selected features		2,3,4,5,6,8,10,12,14,16,23,24, 25,27,29,30,32,33,34,35,36,37,38,39,40

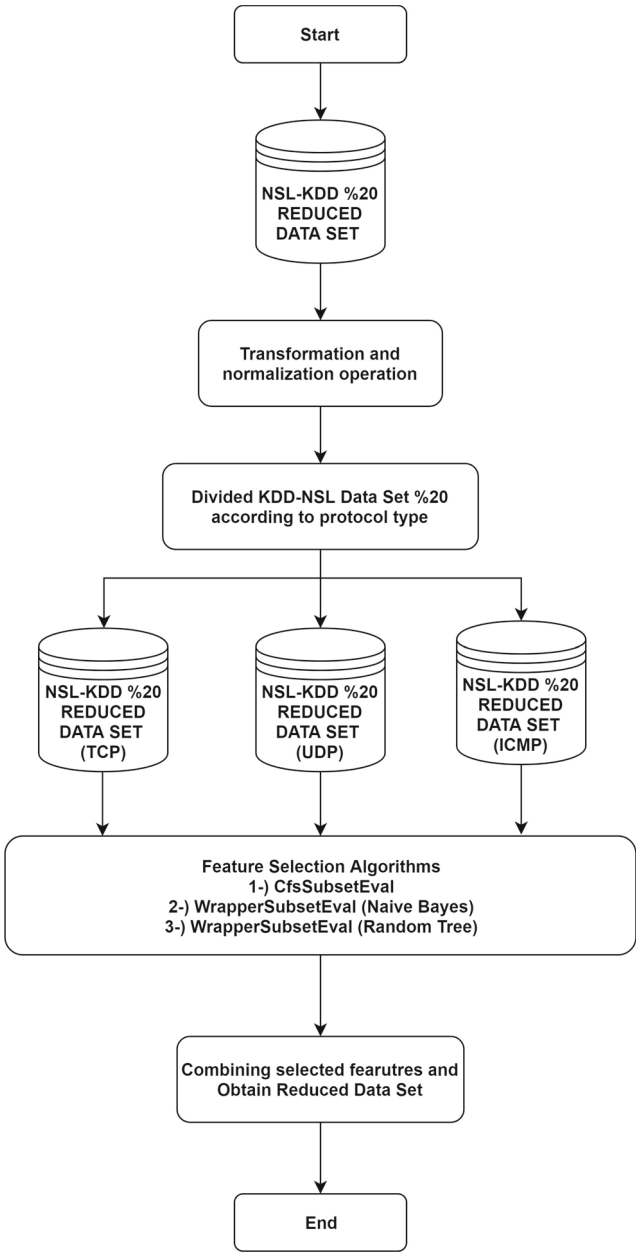

Fig. 3 The block diagram of combining attribute selected different algorithm according to protocol type

datasets according to the type of attack are evaluated as k equals to 10. Algorithms used in determining the pattern of attack type and normal traffic have great importance during training of the system. A single algorithm may be insufficient to generate a pattern of traffic [89]. In order to determine the algorithms with the highest performance and low error rate according to the attack type, the machine learning algortihms have been used singly and in ensemble techniques where algorithms are used together. As a result of the tests performed, it is determined that ensemble techniques achieved quite high performance in some attack types and these methods are used in system design. Since

Table 9 Different feature selection algorithm results according to protocols

Future selection method	Classifier	Protokol type- Selected futures for each method		
		tcp	udp	icmp
CfsSubsetEval	—	3,4,5,11,14,24,30,35,37	3,6,8,30,40	3,5,23,31,32,36
WrapperSubsetEval	Naive Bayes	5,29,30	3,5,8,37,40	5,8,23,31,32
WrapperSubsetEval	Random Tree	3,4,5,6,11,30,35,36,39	2,3,6,8,29,34,35,36,40	3,5,24,32,37,40
Selected features for each protocol		3,4,5,6,11,14,24,29,30,35, 36,37,39	2,3,5,6,8,29,30,34,35, 36,37,40	3,5,8,23,24,31,32, 36,37,40
Overall selected features		2,3,4,5,6,8,11,14,23,24,29,30,31,32,34,35,36,37,38,40		

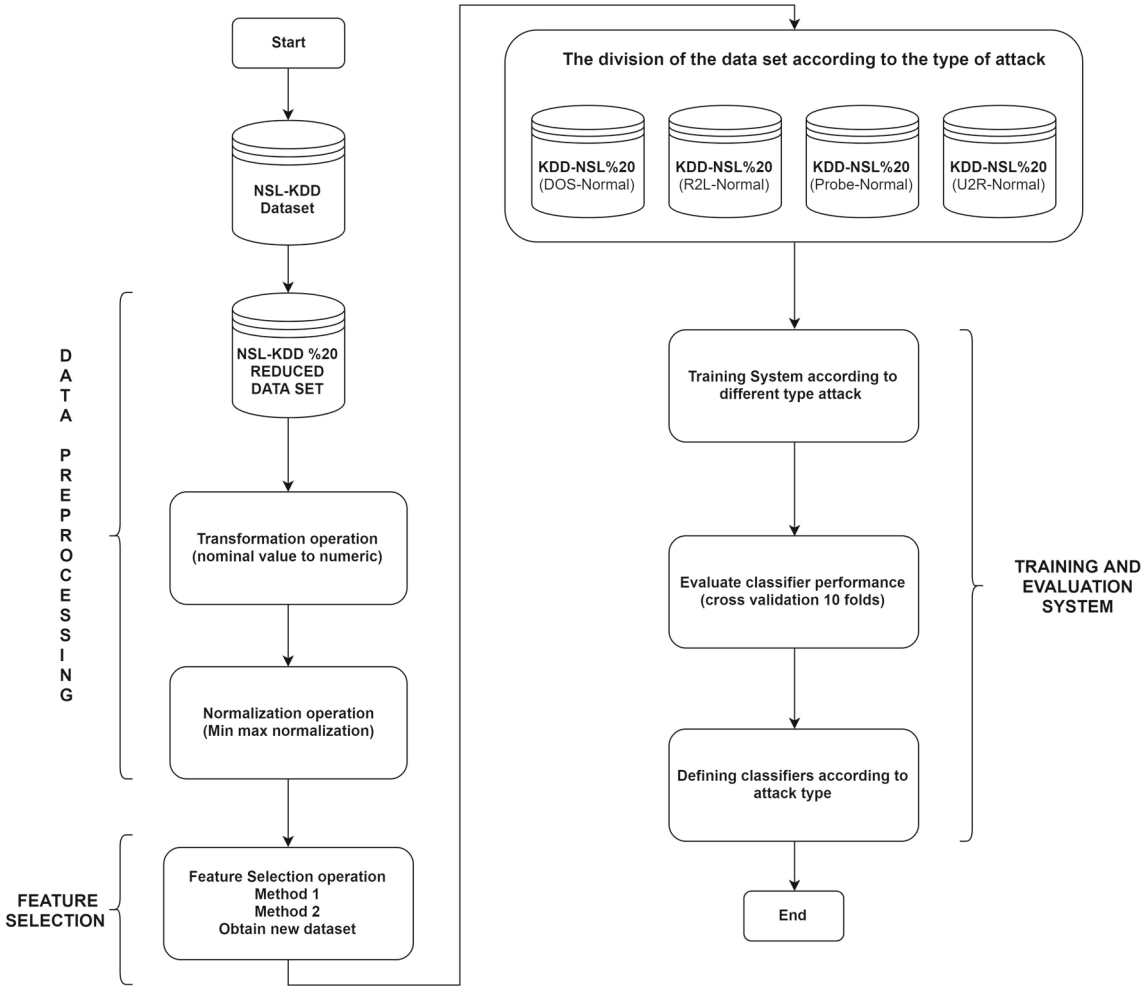

Fig. 4 The block diagram of proposed model

the number of samples belonging to some attack types is few especially in the NSL-KDD %20 training dataset, ensemble method is preferred for detecting the traffic of these attack types. Thanks to the structure in which different algorithms are used, high performance is achieved in attack detection.

After the training and testing processes designed in Fig. 4, the appropriate classification algorithms for the attack type have been determined. In Fig. 5, determined machine learning algorithms according to the attack type and use of the system with new traffic data is seen. When Fig. 5 is examined, the machine learning techniques to be used in system design are determined as follows: RF algorithm for DOS and Probe attacks, stacking method with RF, J48 and Naive Bayes algorithms for R2L attack, J48 and Naive Bayes algortihms for normal-attacks traffic in U2R attacks. After passing the traffic data to be tested from the

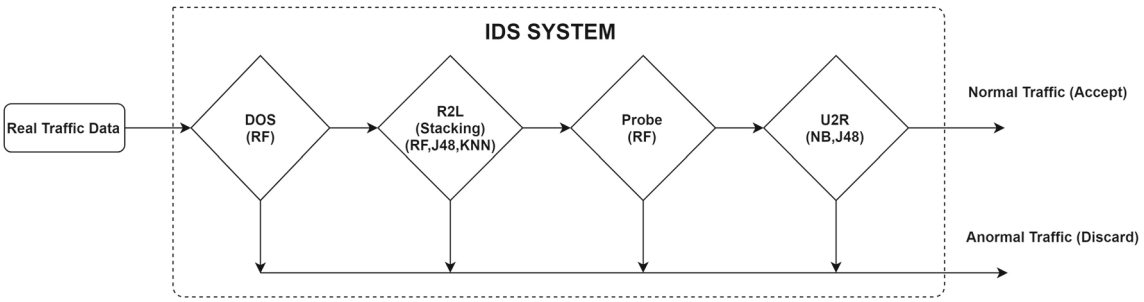

Fig. 5 The layered architecture block diagram for test operation

Table 10 The record number of obtained new datasets

Dataset	The number of attacks instances	The number of normal instances	The total number of instances
Dos+Normal	9234	13449	22683
R2L+ Normal	209	13449	13658
U2R+Normal	11	13449	13460
Probe+Normal	2289	13449	15738

proposed layered system, it is decided whether the traffic belongs to the normal or which type of attack. As a result, traffic data is tested with a high-performance algorithm that is determined to each type of attack, and detection of harmful traffic is performed.

4 The evaluation of the performans tests results

In this section, firstly, the evaluation criteria used for performance evaluation is given. In order to test the performance of the proposed system, performance tests are performed and the results obtained from studies in the literature are compared.

4.1 The evaluation criteria

There are evaluation criteria [90–93] commonly used in the literature to determine the performance of the IDS. In this section, the evaluation criteria used for performance evaluation are explained. The complexity matrix shown in Table 11 is obtained for use in the evaluation criteria. Evaluation criteria are calculated using the values in the complexity matrix. The description of the values in the complexity matrix is as follows:

TP (true-positive): The number of samples that are in the intrusions class in the dataset and are correctly predicted in the intrusions class.
TN (true-negative): The number of samples that are in the normal class in the dataset and are correctly predicted in the normal class.

Table 11 The confusion matrix

		Predicted	
	Intrusions	Normal	
Actual	Intrusions	TP	FN
	Normal	FP	TN

FN (false-negative): The number of samples that are in the intrusions class in the dataset and are incorrectly predicted in the normal class.
FP (false-positive): The number of samples that are in the normal class in the dataset and are incorrectly predicted in the intrusions class.

After the confusion matrix has been obtained, the description of the evaluation criteria calculated using these values is given below.

1. **Accuracy:** Accuracy value is the ratio of the number of samples correctly classified by the system to the total number of samples. It is calculated as seen in (4).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{4}$$

2. **Detection Rate (DR):** The Detection Rate value is the ratio of the TP value to the number of all samples as estimated intrusions .It is calculated as seen in (5).

$$DR = \frac{(TP)}{(TP + FP)} \tag{5}$$

3. **True Positive Rate (TPR):** True Positive Rate is the ratio of the samples classified correctly for a specified class divided by the actual total samples number of that class. It is calculated as seen in (6).

$$TPR = \frac{(TP)}{(TP + FN)} \tag{6}$$

4. **False Positive Rate (FPR):** False Positive Rate is the ratio of the samples classified incorrectly for a specified class divided by the actual total samples number of that class. It is calculated as seen in (7).

$$FPR = \frac{(FP)}{(TN + FP)} \tag{7}$$

5. **F-Measure:** In this evaluation criterion, DR and TPR values are used together to calculate a new value. F-Measure is obtained by calculating the harmonic mean of obtained DR and TPR values. It is calculated as seen in (8).

$$F - Measure = \frac{(2 * DR * TPR)}{(DR + TPR)} \tag{8}$$

6. **Matthews Correlation Coefficients (MCC):** MCC [94] produces more accurate results than other performance criteria, especially on datasets with two classes. It provides the most real result, even on unbalanced distributed datasets. The MCC value is calculated as seen in (9). The calculated value is close to 1 indicating that an accurate classification is made.

$$MCC = \frac{((TP * TN) - (FP * FN))}{\sqrt{(TP+FP) * (TP+FN) * (TN+FP) * (TN+FN)}} \tag{9}$$

7. **Time:** As an evaluation criterion, the processing CPU times are given for tests performed on Weka.

4.2 The performance test results and comparison

In this section, the performance tests of the proposed system are made and the results are evaluated by comparing it with other studies in the literature. Weka [95, 96], an open source data mining tool developed by Waikato University, is used in the performed tests. Weka has many packages to perform operations on datasets. A total of 3 different datasets are used, including the original dataset and obtained by the application of 2 new feature selection methods proposed in the study for the tests.

a. NSL-KDD 20% training+ dataset with 41 features
b. The new dataset with 25 attributes obtained from NSL-KDD 20% training+ dataset using the combining method of different feature selection algorithm
c. The new dataset with 20 attributes obtained from NSL-KDD 20% training+ dataset using the combining method of different feature selection algorithm according to protocol

These three datasets are divided into 4 separate sub datasets according to the attack types. The sample numbers for these datasets are given in Table 10. All types of attack types are tested on this datasets. Evaluation results are shown according to DOS, U2R, Probe and R2L attack types in Tables 12, 13, 14, and 15 respectively. The tables include

Table 12 The perfromance evaluation of DoS attack

Datasets	Total number of instances	Traffic type	Confusion matrix		Accuracy %	DR	TP Rate	FP Rate	F-Measure	MCC	Time (sn)
NSL-KDD %20 (41 attr.)	22683	DOS	9228	6	99.9691	0.9999	0.9993	7.4355E-05	0.999621	0.9993	5.555
		Normal	1	13448							
NSL-KDD %20 cfs+wrapper future selection (25 attr.)	22683	DOS	9231	3	99.9824	0.9999	0.9996	7.4355E-05	0.999783	0.9996	4.340
		Normal	1	13448							
NSL-KDD %20 cfs+wrapper future selection (20 attr.) according to protocol	22683	DOS	9230	4	99.9735	0.9998	0.9995	0.00014871	0.999675	0.9994	4.030
		Normal	2	13447							

Table 13 The perfromance evaluation of U2R attack

Datasets	Total number of Instances	Traffic type	Confusion matrix		Accuracy %	DR	TP Rate	FP Rate	F-Measure	MCC	Time (sn)
NSL-KDD %20 (41 attr.)	13460	U2R	8	3	99.9777	1.0	0.7272	0	0.842105	0.852708	0.26
		Normal	0	13449							
NSL-KDD %20 cfs+wrapper future selection (25 attr.)	13460	U2R	9	2	99.9851	1.0	0.8181	0	0.9	0.9044	0.21
		Normal	0	13449							
NSL-KDD %20 cfs+wrapper future selection (20 attr.) according to protocol	13460	U2R	10	1	99.9926	1.0	0.9090	0	0.9523	0.9534	0.09
		Normal	0	13449							

Table 14 The perfromance evaluation of Probe attack

Datasets	Total number of Instances	Traffic type	Confusion matrix	Accuracy %	DR	TP Rate	FP Rate	F-Measure	MCC	Time (sn)	
NSL-KDD %20 (41 attr.)	15738	Probe	2255	34	99.7331	0.9965	0.9851	0.0005948	0.99077	0.989235	3.86
		Normal	8	13441							
NSL-KDD %20 cfs+wrapper future selection (25 attr.)	15738	Probe	2256	33	99.7331	0.9960	0.9855	0.000669	0.99077	0.98923	2.63
		Normal	9	13440							
NSL-KDD %20 cfs+wrapper future selection (20 attr.) according to protocol	15738	Probe	2259	30	99.7522	0.9960	0.98689	0.000692	0.991443	0.990007	2.36
		Normal	9	13440							

Table 15 The perfromance evaluation of R2L attack

Datasets	Total number of instances	Traffic type	Confusion matrix		Accuracy %	DR	TP Rate	FP Rate	F-Measure	MCC	Time (sn)
NSL-KDD %20 (41 attr.)	13658	R2L	190	19	99.78	0.9453	0.90909	0.000817	0.92682	0.92589	52.86
		Normal	11	13438							
NSL-KDD %20 cfs+wrapper future selection (25 attr.)	13658	R2L	191	18	99.77	0.9363	0.91387	0.00096	0.92493	0.92385	35.32
		Normal	13	13436							
NSL-KDD %20 cfs+wrapper future selection (20 attr.) according to protocol	13658	R2L	187	22	99.77	0.9492	0.89473	0.00074	0.921182	0.920409	34.68
		Normal	10	13439							

the total number of samples and attack type information for the datasets used in the tests. In order to detect each attack, sub-datasets are obtained by combining attack type traffic and normal traffic contents. RF algorithm for DOS and Probe attacks, J48 and NB algorithm for U2R attack type, and stacking method for R2L attack type are used to detect attacks as determined in the previous section. Confusion matrix, Accuracy, DR, TP Rate, FP Rate, F-Measure, MCC and time values are presented for the performance evaluation on the tables.

When Table 12 is examined for DOS attack performance evaluation, it is seen that the highest accuracy and the lowest processing time are obtained in the tests performed on the dataset obtained by using the feature selection algorithms and according to the protocol type. On the other criteria, it is seen that approximately the same and close values are obtained on all datasets. In general, it can be said that the proposed system for DOS attacks has high Accuracy and DR values and low FPR values. When examining the U2R performance data in Table 13, it can be seen that the system used together by J48 and NB produces very successful results in U2R attack detection. In particular, the NB algorithm has been effective in achieving high accuracy and DR rates in detecting U2R traffic. It is determined that the dataset generated by the feature selection according to protocol type has the highest Accuracy, DR, TP Rate, F-Measure and MCC values, lowest FP Rate and processing time. Intrusion detection evaluation results of the probe attack are shown in Table 14. As in other attack types, it is seen that the dataset obtained according to the protocol type in this attack type has the best values in almost all criteria.

It is also evaluated that the performance values are quite good. As can be seen in Table 15, when the performance results of the R2L attack type are examined, it is seen that all datasets are close to each other and good results are obtained. It has been determined that the processing time of the dataset obtained by the feature selection process according to the protocol is less than the processing time of the other datasets.

When the results in Tables 12–15 are evaluated together, it can be said that all attack types performed high performance and attack with low FPR values. The Accuracy values are presented which are the results of tests performed on different datasets according to the attack type in Fig. 6a. Accuracy is one of the most important criteria used to measure the performance of IDS. When the accuracy values are examined, it is determined that the proposed system has a high detection rate of over 99.7% in all attack types. The probe attack type has the lowest rate of attack types with 99.75% attack detection rate. When evaluated in terms of datasets, it is seen that all datasets used have close accuracy values.

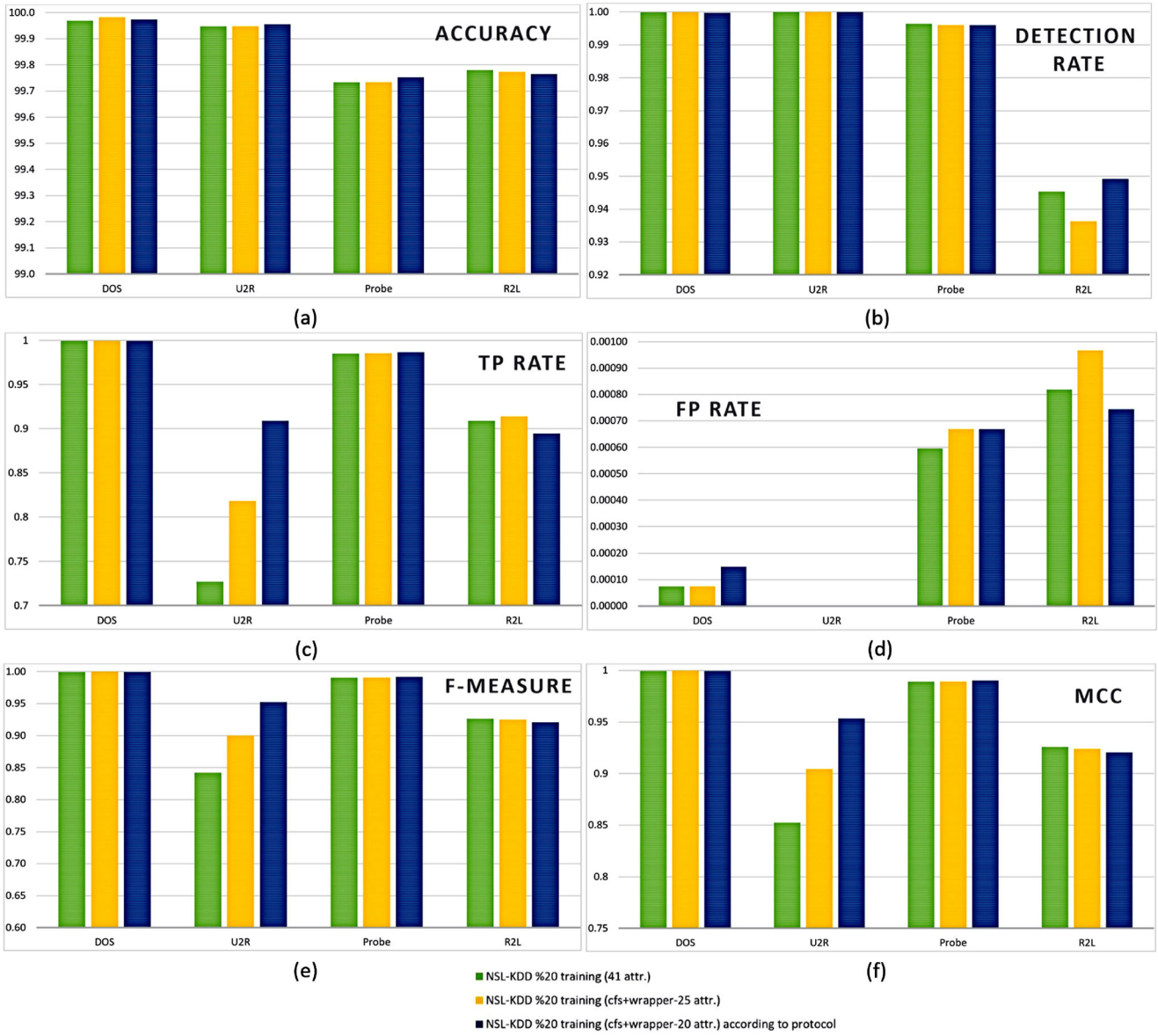

Fig. 6 The result of evaluation criteria according to attack type **a**) Accuracy **b**) Detection Rate **c**) TP Rate **d**) FP Rate **e**) F-Measure **f**) MCC

DR is another important evaluation criterion. Figure 6b shows the DR values obtained according to the attack type. The DR value is calculated close to 1 for DOS and U2R attack types, for all datasets. While the ratio for the probe attack type has a value between 0.99-1, R2L has the highest value with 0.9492 for the attack type. TP Rate and FP Rate are given according to the attack types in Fig. 6c an d. When the TP Rate values on different datasets are examined, it is seen that the TP Rate have the highest value in the DOS attack and 0.98 in the probe attack type. For the U2R and R2L attack, it is seen that the highest TPR values are 0,9 and 0,91 respectively. When FP Rate values are examined,

it is determined that U2R is 0, FP Rate for DOS and Probe attack types is low but R2L attack type is slightly higher than other attack types. However, these FP Rate values are quite acceptable.

Figure 6e shows the F-Measure values according to the attack types. The F-Measure value appears to be approximately 0.99 in all datasets for DOS and Probe attack types. The highest F-Measure values are 0.95 and 0.92 for the U2R and R2L attack types respectively. It can be said that the values obtained for DOS and Probe attacks are fairly good, while for U2R and R2L it is acceptable. It is seen that the MCC values in Fig. 6f have approximate values with F-Measure values.

In Fig. 7, the time values obtained in tests which are another important evaluation criterion are seen as logarithmic. When Fig. 7 is examined, it is seen that on the dataset where the feature selection process is performed according to protocol type, it has less processing time in all attack types. For the U2R attack type, the operation has the time value under one second. Measured processing time for DOS attack detection is about 4 sec. and 2.5 sec. for probe attack. It is clearly seen that the operations on the dataset without feature selection operation are performed longer than the other datasets. For R2L intrusion, it appears that the runtime is increased compared to other attack types, due to the use of the stacking method used by multiple algorithms together and is about 35 seconds.

Table 16 shows the comparison of the performance test results with the proposed system and the studies in the literature. The comparison results of ACC, DR, TPR and FPR which are the most important performance criteria are presented. The evaluation criteria commonly used in literature studies are included in the table and the values that are not used in the literature studies are left blank in the table. The best values are shown as bold according to performance criteria in Table 16. When the Accuracy rates in the DOS attack type are examined, Li et al. and B. Luo, J Xia have reached 100% in their work. It is seen that the test results of the proposed system have better values than all the studies except these studies. It has been found that the DOS and U2R attack types have the lowest FPR value with higher DR and TPR rates than other studies in the literature. The U2R attack type also has the highest accuracy value in the comparative studies. Li et al. and B. Luo, J. Xia's studies have achieved high performance in the DOS type, but accuracy values remain quite below the proposed study for the U2R attack type.

Li et al.'s work has a better value than the proposed system at accuracy value for R2L attack type. Although the DR rate is better than the overall rates, the proposed model has the highest DR rate only for the R2L attack. The highest TPR and the lowest FPR values in the comparative studies for the R2L attack belong to the proposed model. For the probe attack, the best values for all evaluation criteria are obtained by the proposed method. As a result, when the performance values in Table 16 are compared, it is seen that the proposed system has very good performance values in all attack types and has high attack detection capability.

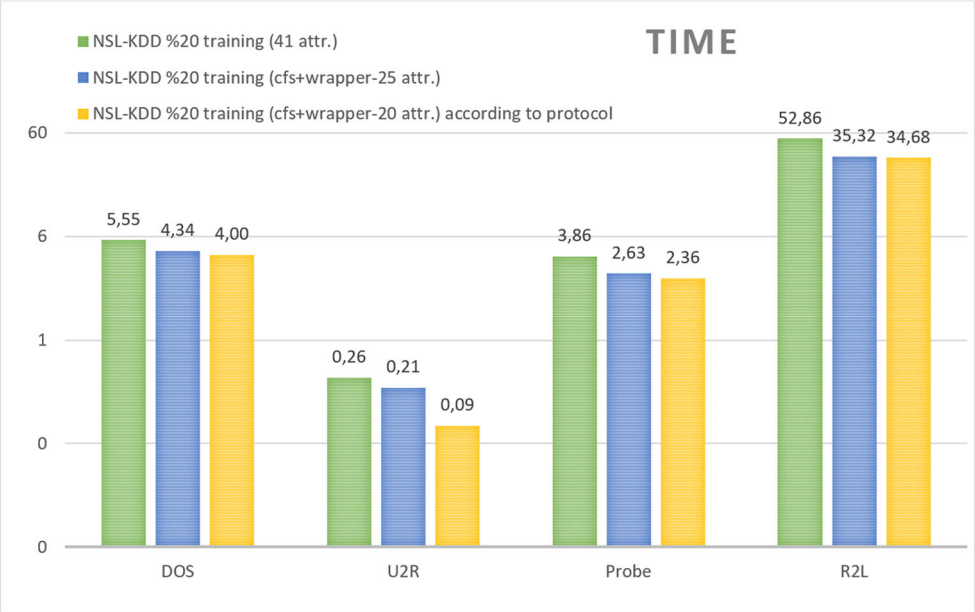

Fig. 7 The results of time evaluaiton according to attack types

Table 16 The Comparison table of proposed model with other methods in the literature

	DOS				U2R				R2L				Probe			
	ACC.	DR	TPR	FPR	ACC.	DR	TPR	FPR	ACC.	DR	TPR	FPR	ACC.	DR	TPR	FPR
Proposed model (NSL-KDD %20- 41 attr.)	99. 9691	**0,9999**	0,9993	0,00074	99.9777	**1**	0,727272	**0**	**99.78**	0.9453	**0,90909**	0.000817	**99.7331**	**0.9965**	0,9851	**0.00059**
Proposed model (NSL-KDD%20 future selection-25 attr.)	**99.9824**	**0,9999**	**0,9996**	**0,000074**	**99.9851**	**1**	**0,8181**	**0**	**99.77**	0.9363	**0,91387**	0.00096	**99.7331**	**0.9960**	0,9855	0.00066
Proposed model (NSL-KDD%20-future selection-20 attr.)	99.9735	**0,9998**	**0,9995**	**0.000148**	**99.9926**	**1**	**0,909090**	**0**	**99.77**	**0.9492**	**0,89473**	**0.00074**	**99.7522**	**0.9960**	**0,9866**	**0.00066**
Li, Y. et al. [37]	**100**	–	–	–	80	–	–	–	**100**	–	–	–	**99.53**	–	–	–
Luo, B., Xia, J. [97]	**100**	–	–	–	80	–	–	–	97	–	–	–	**98.33**	–	–	–
Yao, H. [22]	96,62 (avg)	0,9914	–	–	96,62 (avg)	0,3467	–	–	96,62 (avg)	0.8010	–	–	96.62 (avg)	0,8033	–	–
Singh, R. et al. [12] alpha-ft	97,71 (overall)	98,6 (overall)	0,9918	0,0138	97,71 (overall)	98,6 (overall)	0,5595	0,0010	97,71 (overall)	**98,6** (overall)	0,7701	0,0015	97,71 (overall)	98,6 (overall)	0,9104	0,0039
Singh, R. et al. [12] alpha-fst	97,67 (overall)	98,8 (overall)	0,9913	0,0147	97,67 (overall)	**98,8** (overall)	0,5436	0,0002	97,67 (overall)	**98,8** (overall)	0,7869	0,0017	97,67 (overall)	**98,8** (overall)	0,9048	0,0042
Singh, R. et al. [12] alpha-ft-beta	97,67 (overall)	98,6 (overall)	0,9914	0,0149	97,67 (overall)	98,6 (overall)	0,5675	0,0010	97,67 (overall)	**98,6** (overall)	0,7810	0,0016	97,67 (overall)	98,6 (overall)	0,9035	0,0040
Singh, R. et al. [12]-ANN	94,04 (overall)	–	0,9713	0,0197	94,04 (overall)	–	0	**0**	94,04 (overall)	–	0,7215	0,088	94,04			
(overall)	–	0,6646	0,0027													
Singh, R. et al. [12]-Adaboost	80,55 (overall)	–	0,8941	0,2074	80,55 (overall)	–	–	**0**	80,55 (overall)	–	–	**0**	80,55 (overall)	–	0,6007	0,0215
Singh, R. et al. [12]- NB	82,47 (overall)	–	0,9161	0,022	82,47 (overall)	–	0,8095	0,0507	82,47 (overall)	–	0,8520	0,047	82,47 (overall)	–	0,6950	0,0321
Singh, R. et al. [12]- ELM	92,94 (overall)	–	0,9483	0,0346	92,94 (overall)	–	0	**0**	92,94 (overall)	–	0,6670	0,0022	92,94 (overall)	–	0,7602	0,0153
Guo, C. et al. [11]-ADBCC	–	0,9734	–	–	–	0,8026	–	–	–	0,1151	–	–	–	0,9856	–	–
Guo, C. et al. [11]- Hybrid	–	0,9732	–	–	–	0,7938	–	–	–	0,1285	–	–	–	0,9834	–	–
Lin, W. C. et al. [98]-KNN	99,98	–	–	–	17,31	–	–	–	91,74	–	–	–	**98,49**	–	–	–
Lin, W. C. et al. [98]-CANN	99,68	–	–	–	3,85	–	–	–	57,02	–	–	–	87,61	–	–	–

Table 16 (continued)

	DOS				U2R				R2L				Probe			
	ACC.	DR	TPR	FPR	ACC.	DR	TPR	FPR	ACC.	DR	TPR	FPR	ACC.	DR	TPR	FPR
Lin, W. C. et al. [98]-SVM	82,85	–	–	–	61,54	–	–	–	78,95	–	–	–	96,59	–	–	–
Al-Yaseen, W. L. et al. [24]-ML-SVM	95,57 (overall)	99,57	–	–	95,57 (overall)	16,23	–	–	95,57 (overall)	31,60	–	–	95,57 (overall)	80,94	–	–
Al-Yaseen, W. L. et al. [24]- ML-ELM	93,83 (overall)	96,83	–	–	93,83 (overall)	23,68	–	–	93,83 (overall)	10,14	–	–	93,83 (overall)	84,93	–	–
Al-Yaseen, W. L. et al. [24]-Proposed	95,75 (overall)	99,54	–	–	95,75 (overall)	21,93	–	–	95,75 (overall)	31,39	–	–	95,75 (overall)	87,22	–	–
Liang, H. [99]	–	99,5	–	–	–	14,1	–	–	–	31,5	–	–	–	84,1	–	–
Hoque, M. S. et al. [100]- GA	90 (overall)	99,4	–	–	90 (overall)	18,9	–	–	90 (overall)	5,4	–	–	90 (overall)	71,1	–	–
Horng, S. J. et al. [101]-SVM+BIRCH Clus.	95,7 (overall)	99,5	–	–	95,7 (overall)	19,7	–	–	95,7 (overall)	28,8	–	–	95,7 (overall)	97,5	–	–
Hwang, T. S. et al. [102]-Three Tier IDS	94,7 (overall)	94,3 (overall)	–	0,038 (overall)	94,7 (overall)	94,3 (overall)	–	0,038 (overall)	94,7 (overall)	94,3 (overall)	–	0,038 (overall)	94,7 (overall)	94,3 (overall)	–	0,038 (overall)
Kuang, L. et al. [103]-CSI-KNN	95,1 (overall)	94,6 (overall)	–	0,030 (overall)	95,1 (overall)	94,6 (overall)	–	0,030 (overall)	95,1 (overall)	94,6 (overall)	–	0,030 (overall)	95,1 (overall)	94,6 (overall)	–	0,030 (overall)

5 Conclusion and evaluation results

In this article, a hybrid layered intrusion detection system is proposed using different machine learning techniques according to attack type. The most important contribution of this study is to present a system that performs attack detection with very high performance rates in almost all performance criteria and low error rates when compared with literature studies. In the design of an intrusion detection system, firstly transformation and normalization operations are performed on the dataset. Then, training and test datasets are created using two new methods using different feature selection algorithms. With the proposed feature selection methods, the feature numbers of dataset counts are reduced to almost half and very successful results are obtained with these datasets in the performed tests. The NSL-KDD %20 training dataset is used for the training and testing operations. After the feature selection process, tests are performed to determine the appropriate machine learning technique according to the attack type and the algorithms to be used are determined. The different tests such as Accuracy, DR, TP Rate, FP Rate, F-Measure, MCC and time have been performed to evaluate the performance of the proposed system.

Tables 12–15 presents performance test results according to attack types. Figures 6 and 7 show the results of all attack types graphically based on performance criteria. When the results in the tables are examined, it is seen that all attack types have high accuracy and low FPR values. In the DR and TPR criteria, the best values for the proposed system in the R2L attack type are 0,94 and 0,91 respectively. Compared with the other studies in the literature, it is seen that the values obtained in the DR and TPR criteria are very high in all attack types. In the F-Measure and MCC criteria, the lowest value is found to be 0.92 in the R2L attack type. In other attack types, these criteria have very good values. When the runtimes with respect to the attack types are evaluated, it is determined that the processing time in U2R attack type is very low and R2L attack type is longer than other attack types due to the use of stacking structure. Table 16 shows the performance comparison results of some of the studies in the literature with the presented method. As a result of the comparison with the studies in the literature, it has been found that the proposed system performs attack detection more successfully than many studies in the literature in all attack type.

References

1. Deng R, Zhuang P, Liang H (2017) CCPA: Coordinated Cyber-physical attacks and countermeasures in smart grid. IEEE Trans Smart Grid 8(5):2420–2430
2. Qi L, Dou W, Zhou Y, Yu J, Hu C (2015) A context-aware service evaluation approach over big data for cloud applications. IEEE Transactions on Cloud Computing. https://doi.org/10.1109/TCC.2015.2511764
3. Depren O, Topallar M, Anarim E, Ciliz MK (2005) An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Syst Appl 29(4):713–722
4. Denning DE (1987) An intrusion-detection model. IEEE Trans Softw Eng SE-13(2):222–232
5. Milenkoski A, Vieira M, Kounev S, Avritzer A, Payne BD (2015) Evaluating computer intrusion detection systems: a survey of common practices. ACM Comput Surv (CSUR) 48(1):12
6. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv (CSUR) 41(3):15
7. Ertoz L, Kumar V, Lazarevic A, Srivastava J, Tan PN (2002) Data mining for network intrusion detection. In: Proceedings NSF workshop on next generation data mining, pp 21–30
8. Liao HJ, Lin CHR, Lin YC, Tung KY (2013) Intrusion detection system: a comprehensive review. J Netw Comput Appl 36(1):16–24
9. Wazid M, Das AK (2016) An efficient hybrid anomaly detection scheme using K-means clustering for wireless sensor networks. Wirel Pers Commun 90(4):1971–2000
10. Aljawarneh S, Yassein MB, Aljundi M (2017) An enhanced j48 classification algorithm for the anomaly intrusion detection systems. Clust Comput:1–17. https://doi.org/10.1007/s10586-017-1109-8
11. Guo C, Ping Y, Liu N, Luo SS (2016) A two-level hybrid approach for intrusion detection. Neurocomputing 214:391–400
12. Singh R, Kumar H, Singla RK (2015) An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Expert Syst Appl 42(22):8609–8624
13. Chahal JK, Kaur A (2016) A hybrid approach based on classification and clustering for intrusion detection system. Int J Math Sci Comput 2(4):34–40
14. Saleh AI, Talaat FM, Labib LM (2017) A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. Artif Intell Rev:1–41. https://doi.org/10.1007/s10462-017-9567-1
15. Elbasiony RM, Sallam EA, Eltobely TE, Fahmy MM (2013) A hybrid network intrusion detection framework based on random forests and weighted k-means. Ain Shams Eng J 4(4):753–762
16. Ji SY, Jeong BK, Choi S, Jeong DH (2016) A multi-level intrusion detection method for abnormal network behaviors. J Netw Comput Appl 62:9–17
17. Kim G, Lee S, Kim S (2014) A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Syst Appl 41(4):1690–1700
18. Ravale U, Marathe N, Padiya P (2015) Feature selection based hybrid anomaly intrusion detection system using K means and RBF kernel function. Procedia Comput Sci 45:428–435
19. Laftah Al-Yaseen W, Ali Othman Z, Nazri A, Zakree M (2015) Hybrid modified-means with C4. 5 for intrusion detection systems in Multiagent Systems. The Scientific World Journal
20. Parsaei MR, Rostami SM, Javidan R (2016) A hybrid data mining approach for intrusion detection on imbalanced NSL-KDD dataset. Int J Adv Comput Sci Appl 7(6):20–25

21. Kevric J, Jukic S, Subasi A (2017) An effective combining classifier approach using tree algorithms for network intrusion detection. Neural Comput Appl 28(1):1051–1058
22. Yao H, Wang Q, Wang L, Zhang P, Li M, Liu Y (2017) An intrusion detection framework based on hybrid multi-level data mining. Int J Parallel Prog:1–19. https://doi.org/10.1007/s10766-017-0537-7
23. Farid DM, Zhang L, Rahman CM, Hossain MA, Strachan R (2014) Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. Expert Syst Appl 41(4):1937–1946
24. Al-Yaseen WL, Othman ZA, Nazri MZA (2017) Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. Expert Syst Appl 67:296–303
25. Aslahi-Shahri BM, Rahmani R, Chizari M, Maralani A, Eslami M, Golkar MJ, Ebrahimi A (2016) A hybrid method consisting of GA and SVM for intrusion detection system. Neural Comput Appl 27(6):1669–1676
26. Harb HM, Desuky AS (2011) Adaboost ensemble with genetic algorithm post optimization for intrusion detection. Int J Comput Sci Issues (IJCSI) 8(5):28
27. Kuang F, Xu W, Zhang S (2014) A novel hybrid KPCA and SVM with GA model for intrusion detection. Appl Soft Comput 18:178–184
28. Manickam M, Rajagopalan SP (2018) A hybrid multi-layer intrusion detection system in cloud. Clust Comput:1–9. https://doi.org/10.1007/s10586-018-2557-5
29. Vimala S, Khanaa V, Nalini C (2018) A study on supervised machine learning algorithm to improvise intrusion detection systems for mobile ad hoc networks. Clust Comput:1–10. https://doi.org/10.1007/s10586-018-2686-x
30. Ashfaq RAR, Wang XZ, Huang JZ, Abbas H, He YL (2017) Fuzziness based semi-supervised learning approach for intrusion detection system. Inf Sci 378:484–497
31. Ghosh P, Debnath C, Metia D, Dutta DR (2014) An efficient hybrid multilevel intrusion detection system in cloud environment. IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN, 2278-0661
32. Sangkatsanee P, Wattanapongsakorn N, Charnsripinyo C (2011) Practical real-time intrusion detection using machine learning approaches. Comput Commun 34(18):2227–2235
33. Balamurugan V, Saravanan R (2017) Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation. Clust Comput:1–13. https://doi.org/10.1007/s10586-017-1187-7
34. Benmessahel I, Xie K, Chellal M (2017) A new evolutionary neural networks based on intrusion detection systems using multiverse optimization. Appl Intell 48:2315–2327. https://doi.org/10.1007/s10489-017-1085-y
35. Yang C (2018) Anomaly network traffic detection algorithm based on information entropy measurement under the cloud computing environment. Clust Comput:1–9. https://doi.org/10.1007/s10586-018-1755-5
36. Feng W, Zhang Q, Hu G, Huang JX (2014) Mining network data for intrusion detection through combining SVMs with ant colony networks. Futur Gener Comput Syst 37:127–140
37. Li Y, Xia J, Zhang S, Yan J, Ai X, Dai K (2012) An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Syst Appl 39(1):424–430
38. Wang G, Hao J, Ma J, Huang L (2010) A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. Expert Syst Appl 37(9):6225–6232
39. Wang Y, Feng L (2018) Hybrid feature selection using component co-occurrence based feature relevance measurement. Expert Syst Appl 102:83–99
40. Mukherjee S, Sharma N (2012) Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technol 4:119–128
41. Amiri F, Yousefi MR, Lucas C, Shakery A, Yazdani N (2011) Mutual information-based feature selection for intrusion detection systems. J Netw Comput Appl 34(4):1184–1199
42. Manzoor I, Kumar N (2017) A feature reduced intrusion detection system using ANN classifier. Expert Syst Appl 88:249–257
43. Madbouly AI, Gody AM, Barakat TM (2014) Relevant feature selection model using data mining for intrusion detection system. arXiv:1403.7726
44. Zhang F, Wang D (2013) An effective feature selection approach for network intrusion detection. In: 2013 IEEE eighth international conference on networking, architecture and storage (NAS). IEEE, pp 307–311
45. Pervez MS, Farid DM (2014) Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In: 2014 8th international conference on software, knowledge, information management and applications (SKIMA). IEEE, pp 1–6
46. Ambusaidi MA, He X, Nanda P, Tan Z (2016) Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Trans Comput 65(10):2986–2998
47. Kang SH, Kim KJ (2016) A feature selection approach to find optimal feature subsets for the network intrusion detection system. Clust Comput 19(1):325–333
48. Beulah JR, Punithavathani DS (2018) A hybrid feature selection method for improved detection of Wired/Wireless network intrusions. Wirel Pers Commun 98(2):1853–1869
49. Bhattacharya S, Selvakumar S (2016) Multi-measure multi-weight ranking approach for the identification of the network features for the detection of DoS and Probe attacks. The Comput J 59(6):923–943
50. Bajaj K, Arora A (2013) Dimension reduction in intrusion detection features using discriminative machine learning approach. Int J Comput Sci Issues (IJCSI) 10(4):324
51. Osanaiye O, Cai H, Choo KKR, Dehghantanha A, Xu Z, Dlodlo M (2016) Ensemble-based multi-filter feature selection method for DDos detection in cloud computing. EURASIP J Wirel Commun Netw 2016(1):130
52. Sethuramalingam S, Naganathan ER (2011) Hybrid feature selection for network intrusion. Int J Comput Sci Eng 3(5):1773–1780
53. Sheikhan M, Jadidi Z, Farrokhi A (2012) Intrusion detection using reduced-size RNN based on feature grouping. Neural Comput Appl 21(6):1185–1190
54. De la Hoz E, de la Hoz E, Ortiz A, Ortega J, Martínez-Álvarez A (2014) Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps. Knowl-Based Syst 71:322–338
55. Eesa AS, Orman Z, Brifcani AMA (2015) A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. Expert Syst Appl 42(5):2670–2679
56. Lin SW, Ying KC, Lee CY, Lee ZJ (2012) An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Appl Soft Comput 12(10):3285–3290
57. Online The KDD CUP 1999 Data (1999). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed July 2018
58. Online KDD-NSL Dataset (2009). http://nsl.cs.unb.ca/NSL-KDD/. Accessed July 2018
59. Scott SL (2004) A Bayesian paradigm for designing intrusion detection systems. Comput Stat Data Anal 45(1):69–83
60. Mladenic D, Grobelnik M (1999) Feature selection for unbalanced class distribution and naive bayes. In: ICML, vol 99, pp 258–267

61. Breiman L (2001) Random forests. Mach L 45(1):5–32
62. Alsubhi K, Aib I, Boutaba R (2012) FuzMet: A fuzzy-logic based alert prioritization engine for intrusion detection systems. Int J Netw Manag 22(4):263–284
63. Quinlan RC (1993) 4.5: Programs For machine learning. Morgan Kaufmann Publishers Inc, San Francisco
64. Cannady J (1998) Artificial neural networks for misuse detection. In: National information systems security conference, vol 26, pp 368–381
65. Zhang Z, Shen H (2005) Application of online-training SVMs for real-time intrusion detection with different considerations. Comput Commun 28(12):1428–1442
66. Denoeux T (1995) A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Trans Syst Man Cybern 25(5):804–813
67. Hartigan JA, Wong MA (1979) Algorithm AS 136: a k-means clustering algorithm. J Royal Stat Soc Ser C (Appl Stat) 28(1):100–108
68. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier, New York
69. Alpaydin E (2014) Introduction to machine learning. MIT Press, Cambridge
70. Rodriguez-Galiano VF, Ghimire B, Rogan J, Chica-Olmo M, Rigol-Sanchez JP (2012) An assessment of the effectiveness of a random forest classifier for land-cover classification. ISPRS J Photogramm Remote Sens 67:93–104
71. Malekipirbazari M, Aksakalli V (2015) Risk assessment in social lending via random forests. Expert Syst Appl 42(10):4621–4631
72. Kotsiantis SB, Zaharakis ID, Pintelas PE (2006) Machine learning: a review of classification and combining techniques. Artif Intell Rev 26(3):159–190
73. Sill J, Takács G, Mackey L, Lin D (2009) Feature-weighted linear stacking. arXiv:0911.0460
74. Opitz D, Maclin R (1999) Popular ensemble methods: an empirical study. J Artif Intell Res 11:169–198
75. Wang G, Hao J, Ma J, Jiang H (2011) A comparative assessment of ensemble learning for credit scoring. Expert Syst Appl 38(1):223–230
76. Hall MA, Smith LA (1998) Practical feature subset selection for machine learning. In Computer science'98 proceedings of the 21st Australasian computer science conference ACSC, vol 98, pp 181–191
77. Almuallim H, Dietterich TG (1991) Efficient algorithms for identifying relevant features. In: Proceedings of the 9th Canadian conference on artificial intelligence, pp 38–45
78. Kira K, Rendell LA (1992) The feature selection problem: Traditional methods and a new algorithm. In: AAAI, vol 2, pp 129–134
79. Das S (2001) Filters, wrappers and a boosting-based hybrid for feature selection. In: Icml, vol 1, pp 74–81
80. Liu H, Yu L (2005) Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowl Data Eng 17(4):491–502
81. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28
82. Jantawan B, Tsai CF (2014) A comparison of filter and wrapper approaches with data mining techniques for categorical variables selection. Int J Innov Res Comput Commun Eng 2(6):4501–4508
83. Naseriparsa M, Bidgoli AM, Varaee T (2014) A hybrid feature selection method to improve performance of a group of classification algorithms. arXiv:1403.2372
84. John GH, Kohavi R, Pfleger K (1994) Irrelevant features and the subset selection problem. In: Machine learning proceedings, vol 1994, pp 121–129
85. Chou TS, Yen KK, Luo J (2008) Network intrusion detection design using feature selection of soft computing paradigms. Int J Comput Intell 4(3):196–208
86. Selvakuberan K, Indradevi M, Rajaram R (2008) Combined Feature Selection and classification–A novel approach for the categorization of web pages. J Inf Comput Sci 3(2):083–089
87. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97(1-2):273–324
88. Rodriguez JD, Perez A, Lozano JA (2010) Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE Trans Pattern Anal Mach Intell 32(3):569–575
89. Kittler J, Hatef M, Duin RP, Matas J (1998) On combining classifiers. IEEE Trans Pattern Anal Mach Intell 20(3):226–239
90. Japkowicz N, Shah M (2011) Evaluating learning algorithms: a classification perspective. Cambridge University Press, Cambridge
91. Patil TR, Sherekar SS (2013) Performance analysis of Naive Bayes and J48 classification algorithm for data classification. Int J Comput Sci Appl 6(2):256–261
92. Deng X, Liu Q, Deng Y, Mahadevan S (2016) An improved method to construct basic probability assignment based on the confusion matrix for classification problem. Inf Sci 340:250–261
93. Elshoush HT, Osman IM (2011) Alert correlation in collaborative intelligent intrusion detection systems—A survey. Appl Soft Comput 11(7):4349–4365
94. Liu Y, Cheng J, Yan C, Wu X, Chen F (2015) Research on the Matthews correlation coefficients metrics of personalized recommendation algorithm evaluation. Int J Hybrid Inf Technol 8(1):163–172
95. Online.Weka Data Mining Tool. https://www.cs.waikato.ac.nz/ml/weka/. Accessed July 2018
96. Holmes G, Donkin A, Witten IH (1994) Weka: A machine learning workbench. In: 1994. Proceedings of the 1994 second Australian and New Zealand conference on intelligent information systems. IEEE, pp 357–361
97. Luo B, Xia J (2014) A novel intrusion detection system based on feature generation with visualization strategy. Expert Syst Appl 41(9):4139–4147
98. Lin WC, Ke SW, Tsai CF (2015) CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl-Based Syst 78:13–21
99. Liang H (2014) An improved intrusion detection based on neural network and fuzzy algorithm. J Netw 9(5):1274
100. Hoque MS, Mukit M, Bikas M, Naser A (2012) An implementation of intrusion detection system using genetic algorithm. arXiv:1204.1336
101. Horng SJ, Su MY, Chen YH, Kao TW, Chen RJ, Lai JL, Perkasa CD (2011) A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Syst Appl 38(1):306–313
102. Hwang TS, Lee TJ, Lee YJ (2007) A three-tier IDS via data mining approach. In: Proceedings of the 3rd annual ACM workshop on mining network data. ACM, pp 1–6
103. Kuang L, Zulkernine M (2008) An anomaly intrusion detection method using the CSI-KNN algorithm. In: Proceedings of the 2008 ACM symposium on applied computing. ACM, pp 921–926

Dr. Ünal Çavuşoğlu received his B.Sc. degree in Computer Engineering from Yıldız Technical University, Istanbul, Turkey the M.Sc. degree (in 2014) in Computer Engineering and PhD degree (in 2017) in Computer Engineering from Sakarya University, Sakarya, Turkey. He is currently a researcher in University of Sakarya since 2011. His main research interests are in the fields of computer and communication networks, chaotic system and applications, cryptology and information security.