Future Generation Computer Systems 113 (2020) 418–427

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Genetic convolutional neural network for intrusion detection systems

Minh Tuan Nguyen [a], Kiseon Kim [b,*]

[a] Institute of Research and Development, Duy Tan University, Da Nang 550000, Viet Nam
[b] School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, Republic of Korea

ARTICLE INFO

Article history:
Received 13 October 2019
Received in revised form 6 July 2020
Accepted 16 July 2020
Available online 18 July 2020

Keywords:
Intrusion detection system
Genetic algorithm
Machine learning
Deep learning
Fuzzy C-mean clustering

ABSTRACT

Intrusion detection is the identification of unauthorized access of a computer network. This paper proposes a novel algorithm for a network intrusion detection system (NIDS) using an improved feature subset selected directly by a genetic algorithm (GA)-based exhaustive search and fuzzy C-means clustering (FCM). The algorithm identifies the bagging (BG) classifier and the convolutional neural network (CNN) model as an effective extractor by implementing the GA in combination with 5-fold cross validation (CV) to select the CNN model structure. The deep feature subset extracted by the selected CNN model is put into the BG classifier to validate the performance with the 5-fold CV. The high quality feature set obtained by the three-layered feature construction using the GA, FCM, CNN extractor, and a hybrid CNN and BG learning method significantly improves the final detection performance. Moreover, the highly reliable validation performance results achieved by the 5-fold CV procedure for the proposed algorithm imply a well-fitted application in a practical computer network environment NIDS.

1. Introduction

Identifying unauthorized access of a computer network system is known as intrusion detection. The rationale behind this activity is the security threats and harmful consequences of illegal accesses, which can be instigated by internal or external intruders attempting to obtain valuable computer system information without permission [1]. Moreover, the strong growth of Internet-connected devices, the number of which is anticipated to reach 50 billion by the end of the decade, makes network security attacks become an essentially global problem [2].

Intrusion detection systems (IDS) are novel security mechanisms that use state-of-the-art techniques to secure computer networks against invasions in progress or illegal accesses that have occurred. In addition, cybersecurity work needs resilient and robust design of the IDS, which can withstand large scale ethical hacking and tests in real-time [3]. With respect to protection, an IDS can be categorized as a host-based IDS (HIDS) that uses system log files or a network-based IDS (NIDS) that uses network behaviors. Moreover, there are two forms of these systems, namely, anomaly based and signature-based detection IDSs, which are based on normal and abnormal network patterns, respectively. A signature-based IDS detects an intrusion directly when predefined abnormal network data are detected [4], while an anomaly based IDS analyzes normal network behaviors to determine whether network patterns are considered intrusions [5]. To date, existing IDS designs have been reported with poor performance, including high false positive rates and low accuracy [6–10]. Hence, one of the most important requirements for designing an IDS is improved detection performance.

To improve IDS performance, many studies have focused on IDS design using machine learning (ML) techniques, which rely mainly on supervised or unsupervised methods to learn representative patterns [11]. The most common ML-based algorithms are the naive Bayes, decision tree, support vector machine [12], random forest (RF) [13], and K-nearest neighbor (KNN) [14], Gaussian mixture model [15], and principal component analysis [16]. However, existing effective ML algorithms involve a small amount of input data because the utilizing a very large set of data for the ML method results in a time-consuming process. Moreover, the high-dimension and nonlinear characteristics of large-scale data make them unfit for use in ML methods to solve multiple classification tasks due to decreased performance. Hence, the feature selection utility, which is considered an important requirement of an ML algorithm to remove irrelevant features in the input data and improve the learning process, is inevitable [17].

Recently, deep learning (DL) models, such as long short-term memory, the convolutional neural network (CNN) [18], the deep convolutional generative adversarial network [19], and the deep long-short-term memory recurrent neural network [20], have been widely applied to IDS design. The evolution of the deep features through different layers makes DL capable of extracting a

* Corresponding author.
E-mail address: kskim@gist.ac.kr (K. Kim).

https://doi.org/10.1016/j.future.2020.07.042
0167-739X/

better representative feature subset, enabling better classification performance [21]. Obviously, one of the advantages of DL is that no feature selection requirement is needed. Nevertheless, existing publications show that deep features are learned more efficiently by the DL using a relevant feature subset selected by the feature selection algorithms from the original feature set (OFS) of the input dataset [22]. Clearly, two-layered feature extraction is employed in this strategy, which includes DL as the secondary feature extraction and classification method. Another efficient method to obtain better classification performance for DL in an IDS is the use of additional features, such as a probability score combined with the OFS [23] or an image reshaped from the OFS [24] as the input for different DL algorithms. Additionally, a DL and ML hybrid has been studied intensively in IDS design, in which the former is as the extractor for primary learning to extract a deep feature subset (DFS) from the OFS and the latter plays a secondary learning and classification role [2]. However, the use of DL must be accompanied with careful selection of the structure parameters to achieve an effective model in terms of detection performance improvement. Therefore, the optimization method is essentially necessary to infer a set of parameters for which an optimal DL structure is constructed.

Because (i) most previous studies do not follow a statistically valid process, which results in unreliable IDS application performance in a practical network environment, (ii) the CNN model, which has been carefully investigated and verified to be well-fitted for IDS application design [24,25], has probably not been considered for IDS design in terms of optimal structure, and (iii) IDS classification performance is definitely improved with a feature subset selected by optimization algorithm-based feature selection, in this paper, a novel algorithm using an improved feature subset (IFS) and a CNN and bagging (BG) classifier hybrid algorithm is proposed for a NIDS. First, the IFS is formed as a combination of the features selected carefully by the genetic algorithm (GA) [26] in combination with a KNN fitness function [27] (F-GA) and additional features computed by the fuzzy C-means clustering (FCM) algorithm [28] (F-FCM) from the above F-GA. Second, the best CNN models are selected on the training data using the IFS as the input. Last, the ML classifiers using the DFS extracted by the selected CNN model as the extractor validate the classification performance using 5-fold cross validation (CV). The main contributions of this work are as follows:

- The quality improvement of the deep feature subset, which is then used as the input of the BG classifier, is implemented by three-layered feature construction, consisting of the GA to select the F-GA, FCM to calculate an improved F-FCM, and the CNN model for DFS extraction.
- GA-based optimization method is applied for selection of the most effective CNN structure, which is considered as the deep feature extractor.
- The reliably statistical performance results of various ML and DL models are computed with the 5-folds CV procedure to propose the most productive model for the NIDS.

Obviously, the existing studies consider one-layer feature construction including either the feature set selected by the ML [29–31] or the deep features extracted by the DL [23,24]. Moreover, two-layered feature construction using the DL [22,32–34] or hybrid algorithms of the DL and ML [2,11,35,36] are also implemented in various publications. Compared with previous works, we apply a three-layered feature construction to obtain a DFS with better quality in terms of final classification performance. Indeed, the GA is used for careful selection of the most informative features from the original feature set, which is then reinforced by adding more features calculated by the FCM. This combined feature set, known as IFS, is used as the input of the CNN model for deep feature extraction. Clearly, individual structures of the CNN model produce the DFS with different quality, which show a large impact on the ML classification performance. Furthermore, the GA can blend with DNA cryptography and DL to provide a stronger security measurement and services [37]. Therefore, the GA and 5-folds CV procedure are adopted for selection of the most effective structure of the CNN model as the feature extractor and the ML classifier. The validation performance results imply that the GA is efficient for the CNN structure selection and the proposed NIDS is reliable for practical environment applications.

The rest of the paper is organized as follows: Background explanations for the intrusion problem are described in Section 2, followed by the related work in Section 3. The descriptions of the proposed method are given in Section 4, while Section 5 discusses the evaluation of the method. Sections 6 and 7 present the discussion and conclusion of this study, respectively.

2. Background of the intrusion problem

Information technology plays a crucial role in modern human life. With the rapid growth of the Internet and computer networks, huge amounts of data are exchanged every single moment through the networks, which require the uninterrupted operations of network devices to satisfy end user demands. Unfortunately, device connections to the Internet and computer networks raise cyber-security threats in most daily network activities. Moreover, cyber attacks are now becoming increasingly serious security problems [38,39]. Therefore, identifying various network attacks has become a focus for technicians and experts.

Several traditional methods have been employed for external network security, such as access control, cryptography, and firewalls, which are unable to protect internal attacks. Hence, the IDS is paid intensive attention because it can identify both internal and external intrusion and then provides rapid responses to the attacks. Based on the warnings of the IDS, the intrusion response system, an important component of the IDS, can apply the suitable countermeasures to ensure no interruption of the computer network system monitoring [40]. Among different paradigms of network defense, a NIDS using the latest techniques is designed to overcome the weaknesses of traditional solutions and provide better protection ability for computer networks. In this work, we propose a NIDS design using a combination of ML and DL to reinforce computer network security. Moreover, the performance results are carefully validated to ensure that the proposed NIDS is reliable and applicable to a practical network environment.

3. Related work

ML and DL techniques used for IDS designs in previous studies are given in this section.

A method was developed for selection of training and testing data from a very large set of input data, which are then used as the input of the least square SVM to test the classification performance with different scenarios, such as one against one and one against all [41]. Indeed, this study shows that representative data samples are important for training an effective model in terms of detection performance improvement. Moreover, by training with incremental data, the proposed algorithm is also implemented for a static dataset. The authors of [42] investigate various ensemble classifiers, a decision tree, and a deep neural network (DNN) model using a set of input features, which is selected using a classification and regression tree algorithm. The highest detection performance is achieved by the voting algorithm, which is a combination of five ensemble classifiers. In addition, the proposed algorithm shows an increase in the detection ability for largely imbalanced dataset and consumes short detection delay

in comparison with DNN techniques. Information gain and principal component analysis are combined in the feature selection algorithm to address the most informative feature subset from different databases in [29]. Furthermore, the ensemble classifier shows better performance when using this type of feature subset in comparison with previous studies. In [30], the GA and 10 fold CV procedure are employed to generate input data, feature weights, and parameters for the selection of the optimal SVM model. The effectiveness of the statistical method for selecting the optimal parameters of the proposed model has been proven. Various feature selection algorithms are investigated in [31], such as mutual information, a mutual information firefly algorithm combined with C4.5 and Bayesian network classifiers. Then, a subset of features is selected based on the importance of individual features, which are voted by different feature selection algorithms. Additionally, various ensemble ML models are adopted as the feature selection algorithms to identify the most critical feature subset according to correlation, consistency, information and distance measurement [43]. The optimal feature subset can improve the detection performance and speed the detection process when used as the input for ensemble ML classifiers.

Clearly, DL techniques are suitable for IDS design due to better adaptation to large amounts of intrusion data. Indeed, the recurrent neural network (RNN) model is suggested as the IDS productive algorithm using the multiple layered echo-state machine method for sampling and weight computation [7]. In other words, the optimal RNN structure is selected by an optimization mechanism, which contributes significantly to improved classification performance. Similarly, the GA is adopted to select optimal structure parameters of the deep belief network, as shown in [44]. The authors of [45] propose a distributed artificial neural network structure in which the model is trained separately in the individual network nodes. Moreover, the parameters of each cooperative node are updated for the coordinating master node, which then shares the updated results back to the individual nodes. Obviously, the optimal DL structure plays a vital role in detection performance improvement, as shown in [32], where the effects of depth and number of nodes are investigated to construct the best feed forward DNN using a reduced feature subset selected by the information gain-based feature extraction unit. The performance validation of the feed forward DNN model is implemented with a separated dataset, which results in the higher values than those of the ML models. However, different feed forward DNN models are proposed for binary and multiple detection scenarios, which poses possibly the difficulties for further development in practical environment. A very large-scale dataset is used to estimate the performance of the 5-layer DNN model, which is designed for both NIDS and HIDS [33]. The authors implement an exhaustive search for selection of the structure parameters such as hidden units, learning rate, tanh, and sigmoid activation functions for the optimal DNN model. Certainly, the detection performance of the proposed DNN model is significantly better than that of the ML classifiers, as concluded in this publication. The detection performance of different DL classifiers is measured with a feature subset, which is selected by the sequential feature selection and decision tree algorithm given in [34]. The proposed method has reduced execution time and memory requirements with relatively high classification accuracy. The DNN is applied for the IDS, as shown in [46], where the authors construct a hybrid framework of ML techniques, including the improved genetic algorithm and the simulated annealing algorithm, to find the optimal values of the structure parameters. Another strategy is to use DL models as the extractor in combination with ML classifiers. All input features are used as the input of the DL models to extract a transformed feature set for further classification with ML or DL classifiers. A nonsymmetric deep auto-encoder and RF classifier are proposed as an effective IDS design in [35]. Moreover, the structure of the nonsymmetric deep auto-encoder is also carefully selected to obtain a high-quality deep feature set. In [36], the performance of the stacked sparse auto-encoder is investigated to generate a low-dimension feature subset, which is then used for training an SVM classifier. An improved conditional variational autoencoder is employed to maintain the diversity of the training dataset by learning the sparse representation between input features and classes, reduce the data dimension, and initialize the weight of the DNN hidden layers to achieve global optimization and better detection performance [47]. A self-taught learning framework including sparse autoencoder and SVM is given in [2] for network intrusion detection. Here, the improvement of the classification performance for the shallow and traditional supervised machine learning methods is obtained due to effective dimension reduction of intrusion data and increase in feature representation for both binary and multiple classification scenarios. The authors of [24] suggest a CNN model by investigation of various parameters and conversion of input data into image format. It is proven that the number of computation parameters for the CNN model is reduced by the utility of the image format and the proposed IDS shows better performance than that of traditional intrusion detection methods. A different strategy of model construction is shown in [25], where the cross-layer aggregation network design is employed to deploy an improved CNN model. The authors uses a structure of 2 parallel fully connected layers linked to softmax layer for classification in proposed IDS. In [48], a framework namely cloud computing adoption framework (CCAF) is suggested, which meets the requirements and implementation of multi-layered security. Here, the vulnerabilities of proposed protection system are validated by ethical hacking method, which includes injection of trojans and viruses into the CCAF multi-layered security. A high number of viruses and trojans destroyed or isolated shows that the proposed security framework is extremely reliable for protection of large-scale data.

4. Method

The proposed method is developed with the following 3 phases, as shown in Fig. 1: feature selection, model selection, and model validation. In the first phase, the exhaustive search algorithm, including the GA with a KNN and 5-fold CV fitness function are combined with the FCM to construct an IFS. In the second phase, the CNN structures are selected by the GA using an IFS as the input for the training set and 5-fold CV procedure. In the last phase, the performance of the selected end-to-end CNN models and the DFSs extracted by the selected CNN extractors using the IFS and OFS in combination with different ML classifiers are validated by the 5-fold CV on the validation set.

The 5-fold CV procedure separates the validation dataset into 5 parts; each part is used for evaluation while the remaining parts are used for training. Five repetitions are performed to complete the entire procedure. In this work, the proposed algorithm is first constructed and validated for a multiple classification problem of 5 classes, including 4 network attack types and a normal class. Then, all attack types are considered as the abnormal class, which is used along with the normal class to test the performance of the proposed algorithm. The phases of the method development are described in Sections 4.1, 4.2, and 4.3.

4.1. Feature selection

Most previous studies use all 41 features of the NSL-KDD dataset to construct their proposed IDS. However, the OFS definitely includes irrelevant features, which should be removed to improve the final classification performance of the IDSs [2,11,30–36]. Thus, the GA combined with KNN and FCM are adopted to construct the IFS for further investigation.

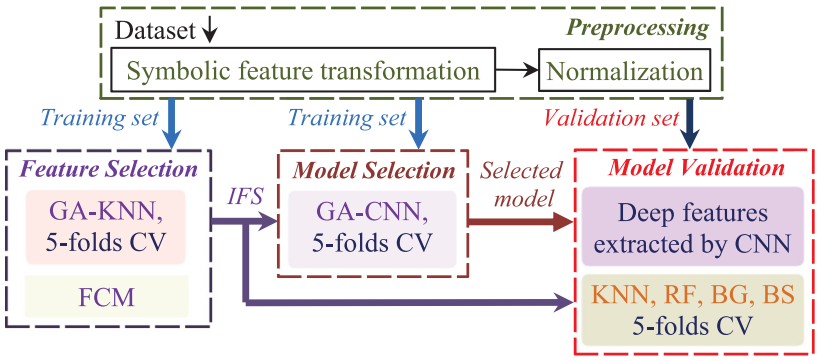

Fig. 1. Block diagram of the method.

4.1.1. Feature search

The individual features in the OFS are estimated by implementing the GA. Here, the chromosome contains a binary, 41-bit, randomly generated string. Each feature is then selected from the OFS based on the bit values of the corresponding chromosome. The feature is chosen or discarded if the bit value is 1 or 0, respectively. The GA contains 30 generations, in which each generation includes 100 chromosomes, known as a population. Moreover, the mutation rate of 0.02, crossover fraction of 0.8, and cloning elements of 2 are used to maintain a stable population diversity and move the superior features to the next generation. The 5-fold CV procedure is applied to select the most productive features based on average minimum fitness values of false positive rate (FPR) for the KNN algorithm [27] on the training set. The GA is repeated 30 times, for which the selected features with high frequency of occurrence are put into the F-GA. KNN is proposed as the fitness function of the GA, in which the distances between each in a set of k records and a test record are determined, and then classified by the majority vote of these distances.

4.1.2. Feature improvement

The FCM [28] algorithm is used to improve the F-GA by calculating additional features (known as F-FCM) from the F-GA. Then, the IFS is a feature combination of all features from F-GA and F-FCM. Here, each record with the F-GA of the entire training set can be grouped into different clusters using FCM. Hence, the membership values ranging from 0 to 1 are used as the F-FCM to represent the probability of a record belonging to a specific cluster.

4.2. Model selection

The CNN model [24] is constructed from the following different layers.

Input layer (IL): The dataset with the CFS is arranged as 1-dimension input data fed into the CNN.

Convolutional layer (CL): This layer plays a role of feature detection at different positions in the input data. The primary element of this layer is the filter, which is a line-shaped object that scans the input data to generate an activation map.

Rectified linear unit layer (ReLu): This layer includes a nonlinear activation function to speed up the learning convergence by mapping nonlinearity into the data.

Maxpooling layer (MP): To reduce computational complexity, the MP layer is used to subsample the feature maps of the preceding CL by computing the maximum responses in small overlapping neighborhoods.

Fully connected layer (FC): The output of previous layer is connected to every unit in the FC, which performs high level inference followed by a nonlinear transformation. This layer can be viewed as a special case of CL with a one-by-one filter size.

Table 1
Details of training and validation datasets.

Dataset	Dos	R2L	U2R	Probe	N	Total
KDDTrain+	45927	995	52	11656	67343	125973
KDDTest+	7458	2754	200	2421	9711	22544
KDDTest-21	4342	2754	200	2402	2152	11850

Output layer (OP): The layer contains the Softmax (SM) layer to encode the probability distribution of a specific class defined by the corresponding unit when the CNN is considered as a classifier and classification layer to assign different labels to the output. The other CNN model parameters, such as regularization of 0.001, momentum of 0.9, and learning rate of 0.1 are preselected as the conventional values [24,25]. Additionally, all the CL and MP use 10 filters with a size of 5 $\times$ 1, while the dimension of the IL is 33 $\times$ 1.

Algorithm 1 : GA for CNN structure selection

(1) Assigning the parameter values: NetS = {1, 2, 3},
Number of blocks = 5;
(2) GA implementation
Population N = 100; Chromosome = {10, 20, 30} of binary bit strings;
$i = 1$;
Repeat
(a) Connection generation based on chromosome;
(b) Construction of CNN;
(c) Division of entire training set into 5 folds $S(j)$;
for j=1 to 5
• Training the CNN model with $S(t)$, $t \neq j$;
• Calculating the mean FPR on $S(j)$;
end
$i = i + 1$;
Until i=N
Computation of the mean FPR of CV;
(3) Selection of CNN structures based on minimum values of FPR.

Firstly, we define a block, which contains a CL and a ReLu. A network segment (NetS) contains 5 consecutive blocks followed by an MP. There are a total of 10 connections between the blocks of a NetS, including 4 connections between the first block and 4 others, 3 connections between the second and 3 others, 2 connections between the third and 2 others, and 1 connection between the last 2 blocks. Secondly, we consider 3 CNN structures; the first structure consists of a NetS with 10 connections of 5 blocks, the second CNN structure includes 2 NetS with 20 connections of 10 blocks, and the third structure contains 3 NetS with 30 connections of 15 blocks. The GA is employed to select the connections between various blocks in individual NetS of the CNN model, as shown in Algorithm 1. There are 3 binary bit strings of 10, 20, and 30 bits generated arbitrarily,

Table 2
The OFS ranked by number of time selection of GA.

Order	Feature name	Number of selection by GA	Order	Feature name	Number of selection by GA
1	duration	30	22	su_attempted	20
2	protocol_type	30	23	wrong_fragment	18
3	service	30	24	hot	18
4	src_bytes	30	25	is_host_login	18
5	dst_bytes	30	26	dst_host_same_srv_rate	18
6	dst_host_serror_rate	29	27	num_compromised	17
7	dst_host_rerror_rate	29	28	serror_rate	16
8	dst_host_same_src_port_rate	28	**	**********	**
9	rerror_rate	27	29	root_shell	14
10	dst_host_srv_serror_rate	26	30	num_outbound_cmds	14
11	flag	24	31	urgent	10
12	srv_serror_rate	24	32	num_shells	9
13	is_guest_login	23	33	logged_in	9
14	num_failed_logins	22	34	count	5
15	dst_host_count	22	35	srv_count	4
16	dst_host_srv_count	21	36	srv_rerror_rate	4
17	num_root	21	37	dst_host_srv_rerror_rate	2
18	num_file_creations	21	38	same_srv_rate	2
19	land	21	39	diff_srv_rate	1
20	num_access_files	21	40	srv_diff_host_rate	0
21	dst_host_diff_srv_rate	21	41	dst_host_srv_diff_host_rate	0

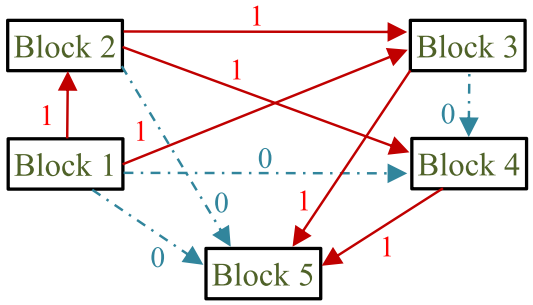

Fig. 2. Connection graph for 5 blocks of a NetS: existing connection with bit 1 and no connection with bit 0.

which represent the connections between 5, 10, and 15 blocks, respectively. A bit value of 1 indicates that a connection exists between 2 blocks, while a bit value of 0 represents no connection. Here, the GA is set for a total of 30 generations, each of which corresponds to a population of 100 chromosomes. The mutation rate, crossover fraction, and cloning elements are assigned as 0.01, 0.8, and 2, respectively. The 5-fold CV procedure is also applied for each CNN structure generated by the GA to estimate the FPR fitness value. Consequently, 3 best CNN structures are selected by the GA corresponding to 1, 2, and 3 NetS (CNN1NetS, CNN2NetS, and CNN3NetS) based on 3 minimum fitness values. A bit string example of 1100110011, which is generated by the GA, representing the connections between the blocks of a NetS is given in Fig. 2.

4.3. Model validation

There are 3 selected CNN models with different numbers of NetS using the IFS estimated on the validation set by the 5-fold CV procedure. Moreover, these CNN models are considered as 3 extractors to generate various DFSs from the last CL for input to 4 ML classifiers, namely, KNN [27], RF, BG, and boosting (BS) [49], and the performance is also validated by the 5-fold CV. The CV procedure is repeated 50 times to compute the mean and standard deviation of the performance. The algorithm with the highest accuracy is proposed and then validated with the OFS to estimate the feature selection performance for multiple classifications. Furthermore, the classification performance of the proposed algorithm using the IFS and OFS is shown for the individual classes. The detection performance of different ML classifiers using the IFS is also validated with 5-fold CV to provide a baseline for comparison with the proposed algorithm.

For the binary classification task, all types of attacks are grouped into an abnormal class. The performance of the proposed algorithm is then evaluated by the 5-fold CV procedure for the IFS, and the OFS is used as the input for the validation set.

5. Evaluation

The dataset description, performance measurement, simulation environment, and results are provided in this section to estimate the performance of the proposed NIDS design algorithm.

5.1. Dataset and preprocessing

The database used for this work is the NSL-KDD dataset, which has been broadly applied for IDS development [50]. This database is the refined version of the KDDCUP99, in which a number of redundant records are eliminated for bias reduction. The NSL-KDD consists of KDDTrain+ used as the training set, and KDDTest+ and KDDTest-21 used as the validation and test sets, respectively. Notably, a few training set attack types are absent in the validation set. A total of 41 features for each record contains 1 to 10 basic features, 11 to 22 content features, and 23 to 41 traffic features. The types of attacks fall into 4 categories: denial of service (DoS), probing (Probe), remote to local (R2L), and user to root (U2R). The remaining data fall into the normal category (N). The details of different attack types and number of records for the training and validation datasets are given in Table 1.

A preprocessing procedure is employed to enhance the dataset quality and the classification performance. A set of 41 features includes 38 numeric and 3 nonnumeric features, (protocol type, service, and flag), which are converted into numeric values by the probability density algorithm. A few features have a large distance between their minimum and maximum values, such as 0 and 58329 for the duration feature, 0 and 1.3×10^9 for the src_bytes and dst_bytes features. Therefore, logarithmic scaling is applied to reduce such distances, which are then scaled with min–max normalization into a [0,1] range.

Table 3
Validation performance of the CNN model and the CNN extractor in combination with ML classifiers on KDDTest+.

Model	Classifier	Ac (%)	FPR (%)	TPR (%)
CNN1NetS		95.42 ± 4.62	0.94 ± 0.08	88.44 ± 4.33
CNN1NetS extractor	KNN	96.98 ± 2.75	0.75 ± 0.05	90.94 ± 2.47
	RF	97.42 ± 4.35	0.57 ± 0.02	93.52 ± 4.24
	BG	**98.24 ± 0.06**	**0.52 ± 0.01**	**95.44 ± 0.46**
	BS	97.04 ± 2.81	0.63 ± 0.19	96.72 ± 2.99
CNN2NetS		86.58 ± 13.15	0.92 ± 0.17	79.08 ± 12.06
CNN2NetS extractor	KNN	95.56 ± 5.31	0.79 ± 0.06	89.38 ± 5.05
	RF	94.21 ± 8.15	0.51 ± 0.05	89.75 ± 7.79
	BG	96.97 ± 3.87	0.65 ± 0.04	92.00 ± 3.71
	BS	93.34 ± 7.51	0.73 ± 0.07	93.11 ± 7.46
CNN3NetS		92.81 ± 7.35	1.07 ± 0.13	83.88 ± 6.76
CNN3NetS extractor	KNN	95.55 ± 5.33	0.81 ± 0.06	89.44 ± 5.00
	RF	96.91 ± 4.73	0.57 ± 0.11	91.66 ± 4.67
	BG	96.02 ± 5.95	0.58 ± 0.04	91.32 ± 5.70
	BS	95.22 ± 5.87	0.67 ± 0.07	94.73 ± 5.90

Table 4
Validation performance of the proposed algorithm for individual classes on KDDTest+.

Class	IFS		OFS	
	Ac (%)	FPR (%)	Ac (%)	FPR (%)
Normal	98.61 ± 0.07	1.53 ± 0.06	96.45 ± 6.17	1.49 ± 0.16
Dos	99.56 ± 0.05	0.22 ± 0.03	97.60 ± 6.27	0.16 ± 0.03
Probe	97.85 ± 0.23	0.23 ± 0.02	96.17 ± 6.31	0.22 ± 0.04
R2L	94.96 ± 0.21	0.54 ± 0.03	93.09 ± 5.90	0.58 ± 0.06
U2r	81.20 ± 2.26	0.06 ± 0.01	81.76 ± 6.52	0.08 ± 0.02

Table 5
Validation performance of the proposed algorithm for multiple and binary classifications on KDDTest+.

Multiple classification			
Feature set	Ac (%)	FPR (%)	TPR (%)
IFS	98.24 ± 0.06	0.52 ± 0.01	95.44 ± 0.46
OFS	96.26 ± 0.06	0.51 ± 0.05	93.01 ± 0.06
Binary classification			
Feature set	Ac (%)	FPR (%)	TPR (%)
IFS	97.94 ± 2.99	1.33 ± 0.08	97.49 ± 3.85
OFS	97.44 ± 4.55	1.35 ± 0.06	97.16 ± 1.11

Table 6
Confusion matrix of the proposed algorithm on KDDTest-21.

True	Predicted				
	N	DoS	Probe	R2L	U2R
N	2074	9	18	51	0
DoS	12	4323	7	0	0
Probe	17	9	2376	0	0
R2L	55	3	8	2685	3
U2R	19	2	2	11	166

Ac of 98.09%, FPR of 0.49%, TPR of 95.07%.

5.2. Performance measurement

The performance of the proposed method is measured with 3 parameters: accuracy (Ac), true positive rate (TPR), and false positive rate (FPR). The most important performance metric is Ac, which is computed as the proportion of records that are correctly classified. The TPR and FPR show the proportion of records identified correctly and incorrectly, respectively. Notably, the Ac of each class is computed as the TPR of the corresponding class.

$$Ac = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$TPR = \frac{TP}{TP + FN} \quad (2)$$

$$FPR = \frac{FP}{FP + TN} \quad (3)$$

While the true positive (TP) and false negative (FN) values give the number of attack records classified correctly and incorrectly into the attack and the normal classes; the true negative (TN) and false positive (FP) values represent the number of normal records identified correctly and incorrectly as the normal and the attack classes, respectively.

5.3. Simulation environment

The simulation is implemented on a 4.1 GHz Intel Core i5 processor personal computer with 16 Gb of memory, a GTX 1050 graphics card and MATLAB 2018a.

5.4. Simulation results

5.4.1. Feature selection

The original features are ranked according to the number of times they are selected by the GA, as shown in Table 2. The features are put into the F-GA if they are selected in more than 50% of the total repetitions. Consequently, a total of 28 features selected over 15 times are considered for further steps. The FCM uses the F-GA to calculate the F-FCM of 5 features as the additional features. The IFS is a combination of F-GA and F-FCM including 33 features.

5.4.2. Model selection

Fig. 3 shows the CNN model structures selected by the GA, corresponding to 1 NetS, 2 NetS, and 3 NetS. These selected CNN models are also used as the extractors to generate the DFS for the input to the different ML classifiers.

5.4.3. Model validation

Table 3 presents the validation performance results of the CNN models and extractors in combination with various ML classifiers on the validation set. The highest accuracy is 98.24% for the CNN1NetS considered as the extractor and BG classifier using the IFS, which are selected as the proposed algorithm for the IDS in this work.

The validation performance results for the proposed algorithm for individual classes and multiple, binary classifications are given in Tables 4 and 5. The confusion matrix for the test set for the

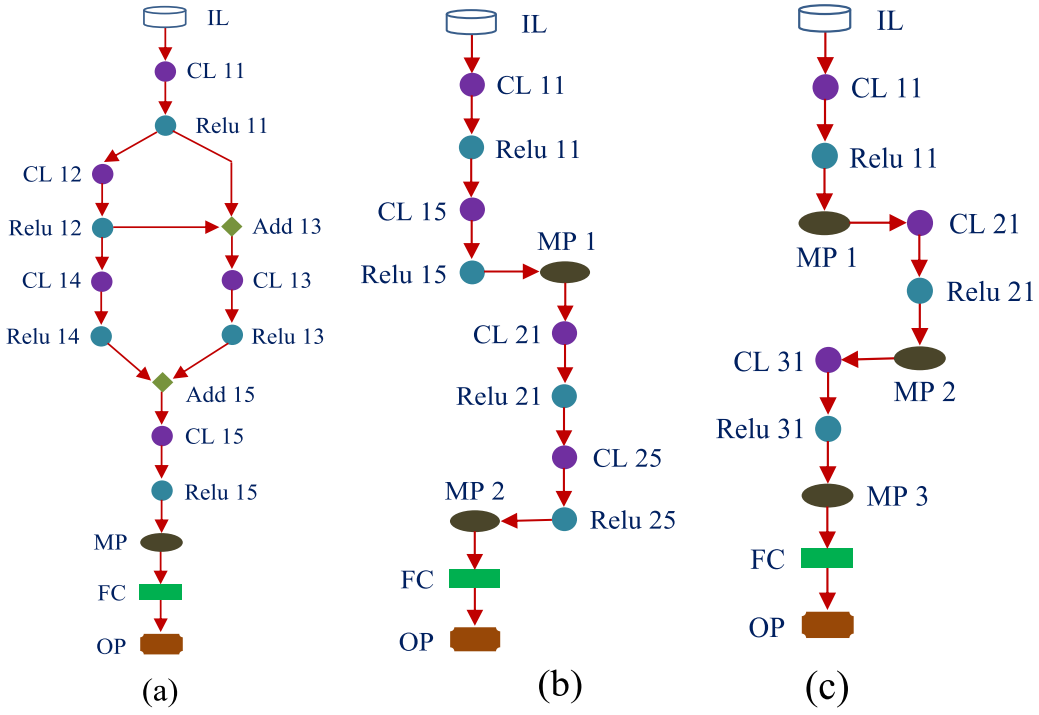

Fig. 3. Selected CNN models by the GA: (a) CNN1NetS, (b) CNN2NetS, (c) CNN3NetS.

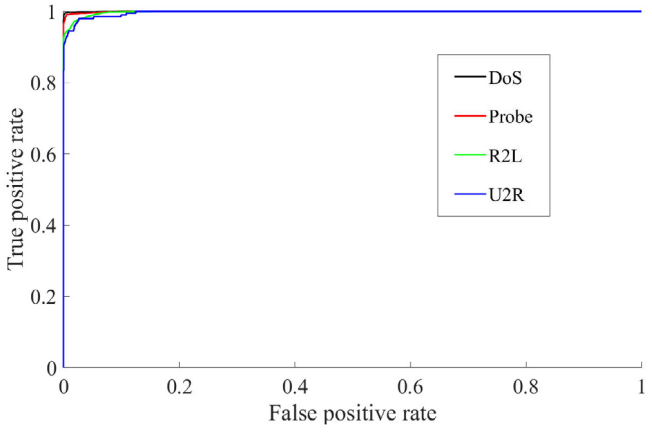

Fig. 4. ROC curves of individual attack types and normal class.

proposed algorithm using the IFS is shown in Table 6. Fig. 4 shows the receiver operating characteristic (ROC) curves of individual attack types and the normal class with areas under the curves (AUC) of 99.9%, 99.8%, 99.8%, and 99.7% for the DOS, probe, R2L, and U2R attack classes, respectively. The validation performance of the ML classifiers is given in Table 7.

6. Discussion

This study develops an effective algorithm that is applicable to a practical IDS and plays a vital role in detecting illegal activities over computer networks. Although different state-of-the-art methods have been employed for multiple intrusion problem classification tasks, their overall performance is still relatively poor, which makes these IDS designs difficult to apply for practical network environments.

The DL technique, which has been considered for IDS design [7,21,23–25,44,45], can be viewed as having two parts: feature extraction and classification. The first part plays the DFS extraction role learned through multiple DL layers using the OFS, which is then put into the second part for classification in these studies. The disadvantage of this method is that the DFS quality is possibly reduced due to being generated directly from the OFS, which certainly includes irrelevant features. Thus, another two-layered feature construction strategy is introduced in [2,11,35,36], including the feature selection to select the relevant features from the OFS for the DL input. Compared to the previously mentioned studies, my method proposes three-layered feature construction to increase the feature quality, which is definitely effective in improving the final classification performance. Indeed, the GA-based feature selection plays a role of the first layer to select the most informative feature subset, which is then improved by adding more efficient features computed by the second layer of feature improvement using FCM. The third layer is the CNN model for DFS feature extraction, which yields significantly enhanced feature quality. As shown in Table 3, the validation performance of the CNN extractors in combination with ML classifiers is higher than that of the end-to-end CNN models using the IFS, which is considered as the two-layered feature construction. Therefore, the CNN model is an effective IDS application as the extractor combined with an ML classifier. Furthermore, the proposed algorithm using the IFS produces better performance than that using the OFS, as shown in Tables 4 and 5. Obviously, the ML classifiers' validation performance shown in Table 7 are relatively low compared to the proposed algorithm. These results imply that the GA and FCM are effective in constructing an informative IFS. The rationale behind the use of the CL for DFS extraction is that the feature mapping is implemented in the CL to convert the input features into a high-dimension feature set with better information and diversity. Moreover, other layers, such as the ReLu, MP, and FC, perform the linear activation and subsampling functions, which reduce the nonlinearity of the DFS extracted by the preceding CL. As a result, the final detection performance is decreased due to reduction of the DFS nonlinearity, also known as reduction of DFS diversity.

Another important characteristic is that the parallel structure of the improved CNN model [25] yields better performance than

Table 7
Validation performance of ML classifiers on KDDTest+.

Classifier	IFS			OFS		
	Ac (%)	FPR (%)	TPR (%)	Ac (%)	FPR (%)	TPR (%)
KNN	94.24 ± 0.06	1.59 ± 0.02	85.63 ± 0.18	93.71 ± 0.07	1.73 ± 0.03	84.19 ± 0.83
RF	96.22 ± 0.05	1.08 ± 0.02	83.58 ± 0.33	92.67 ± 7.90	0.95 ± 0.08	78.67 ± 6.70
BG	95.33 ± 0.08	1.32 ± 0.03	83.98 ± 0.42	90.69 ± 8.48	1.24 ± 0.11	78.57 ± 7.33
BS	95.62 ± 0.07	1.03 ± 0.02	95.30 ± 0.20	94.40 ± 0.11	1.31 ± 0.02	95.04 ± 0.24

Table 8
Overall performance comparison of proposed method with existing algorithms for multiple classification.

Ref., Year	Method	Number of selected features	Ac (%)	FPR (%)	TPR (%)	Cons	Pros
[42], 2019	- Classification and regression tree based feature selection - Voting algorithm of 5 ML classifiers	17	85.2	NA	85.2	- Preselection of feature number. - Manually setting up voting weight based on adjusting proportion of training data	- Ensemble learning of 4 models to gather advantages of different algorithms - Short processing time - Detection improvement for large-scale imbalanced data
[24], 2018	- Conversion of data into images - CNN	NA	79.5	27.9	68.7	- Only one CNN structure - Low accuracy - Use of all input features	- Reduction of number of computation parameters by the use of image data format - Effective cost function weight coefficients of the classes - Comparison with RNN and ML classifiers
[25], 2019	Improved CNN	NA	95.4	0.76	95.6	- Only CNN model - Use of all input features	CNN structure with cross layer converged by stochastic gradient descent algorithm
[34], 2019	- Sequential feature selection and decision tree based feature selection - Long short-term memory	12	92.0	NA	NA	- Separated feature sets for different datasets - Investigation only for DL algorithms	- Measurement of memory profiler of the proposed algorithm - Reduction of execution time and memory amount
[32], 2019	- Information gain based feature extraction unit as feature selection - Feed forward DNN	21	86.2	NA	NA	- Implementation of model with small number of hidden layers and nodes - Comparison with only ML algorithms	Impact of depth and number of neurons of feed forward DNN on accuracy
[33], 2019	5-layers DNN	NA	78.5	NA	78.5	- Use of all input features - Consideration of small set of DNN parameters - Low accuracy	- DNN 5 layers for both HIDS and NIDS - Model estimation on large-scale datasets - Comparison with different ML models
[2], 2018	- Sparse autoencoder for reduction of feature dimension - SVM	NA	80.5	NA	68.3	- Use of all input features - Implementation of one algorithm	Effective sparse autoencoder for feature dimension reduction and self-taught learning framework for IDS
Proposed algorithm	- GA and KNN based feature selection - FCM for feature improvement - CNN for DFS extraction - BG classifier	33	98.2	0.5	95.4	Time consuming for proposed method	- Three-layered feature construction - GA or selection of optimal CNN - Validation method - Comparison with various ML classifiers

that of the conventional CNN model consisting of a single input and output for each layer [24]. Basically, a cross-layer approach is adopted for the parallel CNN model, in which the output of the first convolutional layer is used as the input of different following layers. In other words, the layer information can be reused effectively by the others, which contributes to the final performance improvement. However, parallel structures of the CNN models are preselected for the IDS designs in previous researches, which lead to omits possibly effective cross-layer CNN models. Therefore, the GA-based exhaustive search is deployed in this work to select the best cross-layer structure for the CNN model. As a result, the CNN 1NetS, which is selected by the GA, in combination with the BG classifier, has a relatively high detection performance. The GA is extremely suitable for selecting cross-layer CNN structures.

Avoiding the overfitting problem is an essential requirement for training ML and DL models because the performance of the trained models is significantly reduced on the different datasets. To overcome this problem, the 5-fold CV-based statistical method has been largely used to construct ML and DL algorithms in various research areas. Even so, most publications related to intrusion detection tasks have ignored it during IDS algorithm development. Hence, the 5-fold CV is implemented in this study to achieve statistical performance results, which lead to a final algorithm with high reliability. The validation dataset is divided randomly into 5 folds of records. The training data consists of 4 folds, while the last fold is the testing data. This process is run 5 times to complete an entire procedure for calculating average performance, and every fold is used as the testing data. The 5-fold CV is arbitrarily repeated 50 times for all the individual models, the mean and standard deviation of the performance are computed again to address the most stable CNN model. Thus, the validation performance results are significantly reliable. The small standard deviation values of Ac (0.06%), FPR (0.01%), and TPR (0.46%) for the proposed algorithm including the CNN 1NetS extractor and BG classifier mean that the extractor remains highly stable compared with the others, as shown in Table 3.

The hybrid learning method of the CNN extractor and BG classifier in the proposed algorithm is basically formed by replacing the classification part of the CNN by the BG classifier. Obviously, the learning process and classification are available for the BG classifier, while the classification part of the CNN does not involve a learning process. Consequently, the second learning process definitely contributes to the improved final detection performance. The proposed algorithm yields better performance compared with existing methods for multiple classifications, as shown in Table 8. Notably, the latest publications employing the NSL-KDD dataset, which is also used in this research, are collected for the performance comparison.

The limitation of this research is the time consumption, due to implementing a GA-based exhaustive search method to select a relevant feature subset and an effective CNN structure. Moreover, repetition of the 5-fold CV procedure requires also a very large amount of time spent in practical simulation. Additionally, the input data are likely imbalanced, which make possibly the proposed algorithm biased toward the predominated class when being applied in practical environment of computer networks. Subsequently, validation of the proposed algorithm with diverse intrusion databases, especially for the finite size of data, is open for the further research.

7. Conclusion

Correct IDS classification of dangerous computer network attacks is important for rapid responses to these attacks in terms of network security. In this work, we propose an efficient algorithm for deployment in a practical NIDS, using an IFS and a combination of a CNN extractor and a BG classifier. The IFS was carefully constructed by the GA to select the most informative feature set from the OFS and the FCM to compute additional features. Moreover, the IFS was then used as the input to the CNN model, known as an extractor, which was selected among various CNN structures generated by the GA. As a result, a high-quality DFS was produced by the extractor, which learned the most representative characteristics of the attack types through multiple layers. The BG classifier performance was then validated by the 5-fold CV procedure using the DFS as the input. Clearly, three-layer feature construction, which includes the GA-based feature selection, FCM base feature improvement, and CNN model-based deep feature extraction, was extremely effective in finding the most representative feature subset and improve the final detection performance. Furthermore, the 5-fold CV applied for all development phases of the proposed method avoided the overfitting problem and achieved reliable evaluation performance results for the proposed IDS design algorithm. In addition, the robust learning of the hybrid learning method containing the CNN model as an extractor and the BG classifier helped improve the final classification performance of the proposed algorithm.

The simulation showed the successful utility of hybrid algorithm for cybersecurity. Indeed, we proposed an effective method to make massive intrusion data more separable by the use of DL technique as the feature extractor along with ML classifier. As a result, we expect that deployment of the proposed algorithm over the practical internet systems would improve the computer network security against the illegal activities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] B.B. Zarpelão, R.S. Mianib, C.T. Kawakania, S.C. de Alvarenga, A survey of intrusion detection in Internet of Things, J. Netw. Comput. Appl. 84 (2017) 25–37.
[2] M. Al-Quatf, Y. Lasheng, M. Al-Habib, K. Al-Sabahi, Deep learning approach combining sparse autoencoder with SVM for network intrusion detection, IEEE Access 6 (2018) 52843–52856.
[3] A.S. Sohal, R. Sandhu, S.K. Sood, V. Chang, A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments, Comput. Secur. 74 (2018) 340–354.
[4] W. Kong, J. Shen, P. Vijayakumar, Y. Cho, V. Chang, A practical group blind signature scheme for privacy protection in smart grid, J. Parallel Distrib. Comput. 136 (2020) 29–39.
[5] J. Ren, J. Guo, W. Qian, H. Yuan, X. Hao, H. Jingjing, Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms, Secur. Commun. Netw. 2019 (2019) 1–11, Article ID 7130868.
[6] P. Li, W.H. Zhou, Hybrid intrusion detection algorithm based on k-means and decision tree, Comput. Mod. 37 (2019) 12–16.
[7] T.A. Tchakoucht, M. Ezziyyani, Multilayered echo-state machine: A novel architecture for efficient intrusion detection, IEEE Access 6 (2018) 72458–72468.
[8] R. Yahaloma, A. Sterena, Y. Nameria, M. Roytmana, A. Porgadorb, Y. Elovici, Improving the effectiveness of intrusion detection systems for hierarchical data, Knowl.-Based Syst. 168 (2019) 59–69.
[9] M. Ahsana, M. Mashuria, M.H. Leeb, H. Kuswantoa, D.D. Prastyo, Robust adaptive multivariate hotelling's T2 control chart based on kernel density estimation for intrusion detection system, Expert Syst. Appl. 145 (2020) http://dx.doi.org/10.1016/j.eswa.2019.113105.
[10] J. Lee, J. Kim, I. Kim, K. Han, Cyber threat detection based on artificial neural networks using event profiles, IEEE Access 7 (2019) 165607–165626.
[11] N. Marir, H. Wang, G. Feng, B. Li, M. Jia, Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark, IEEE Access 6 (2018) 59657–59671.
[12] A.L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Commun. Surv. Tutor. 18 (2016) 1153–1176.
[13] N. Farnaaz, M.A. Jabbar, Random forest modeling for network intrusion detection system, Procedia Comput. Sci. 89 (2016) 213–217.
[14] A.A. Aburomman, M.B.I. Reaz, A novel SVM-kNN-PSO ensemble method for intrusion detection system, Appl. Soft Comput. 38 (2016) 360–372.
[15] J. Liu, W. Zhang, Z. Tang, Y. Xie, T. Ma, J. Zhang, G. Zhang, J.P. Niyoyita, Adaptive intrusion detection via GA-GOGMM-based pattern learning with fuzzy rough set-based attribute selection, Expert Syst. Appl. 139 (2020) 1–17.

[16] J. Camacho, R. Therón, J.M. García-Giménez, G. Maciá-Fernández, P. García-Teodoro, Group-wise principal component analysis for exploratory intrusion detection, IEEE Access 7 (2019) 113081–113093.
[17] I. Guyon, S. Gunn, M. Nikravesh, L.A. Zadeh, Feature Extraction Foundations and Application, Springer Berlin Heidelberg, NY, USA, 2006.
[18] A. Liu, B. Sun, An intrusion detection system based on a quantitative model of interaction mode between ports, IEEE Access 7 (2019) 161725–161740.
[19] J. Yang, T. Li, G. Liang, W. He, Y. Zhao, A simple recurrent unit model based intrusion detection system with DCGAN, IEEE Access 7 (2019) 83286–83296.
[20] S.M. Kasongo, Y. Sun, A deep long short-term memory based classifier for wireless intrusion detection system, ICT Express, Available online 22 August 2019.
[21] C. Yin, Y. Zhu, J. Fei, X. He, A deep learning approach for intrusion detection using recurrent neural networks, IEEE Access 5 (2017) 21954–21961.
[22] N. Chouhan, A. Khan, H.R. Khan, Network anomaly detection using channel boosted and residual learning based deep convolutional neural network, Appl. Soft Comput. 83 (2019) Article ID 105612.
[23] F.A. Khan, A. Gumaei, A. Derhab, A. Hussain, A novel two-stage deep learning model for efficient network intrusion detection, IEEE Access 7 (2019) 30373–30385.
[24] K. Wu, Z. Chen, W. Li, A novel intrusion detection model for a massive network using convolutional neural networks, IEEE Access 6 (2018) 50850–50859.
[25] H. Yang, F. Wang, Wireless network intrusion detection based on improved convolutional neural network, IEEE Access 7 (2019) 64366–64374.
[26] R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, John Wiley & Sons, NJ, USA, 2004.
[27] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, NJ, USA, 2001.
[28] S. Miyamoto, H. Ichihashi, K. Honda, Algorithms for Fuzzy Clustering, Springer-Verlag Berlin Heidelberg, Germany, 2008.
[29] F. Salo, A.B. Nassif, A. Essex, Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection, Secur. Commun. Netw. 2018 (2018) 1–11, Article ID 1914980.
[30] P. Tao, Z. Sun, Z. Sun, An improved intrusion detection algorithm based on GA and SVM, IEEE Access 6 (2018) 13624–13631.
[31] B. Selvakumar, K. Muneeswaran, Firefly algorithm based feature selection for network intrusion detection, Comput. Secur. 81 (2019) 148–155.
[32] S.M. Kasongo, Y. Sun, A deep learning method with filter based feature engineering for wireless intrusion detection system, IEEE Access 7 (2019) 38597–38607.
[33] R. Vinayakumar, M. Alazab, K.P. Soman, P. Poornachandran, A. Al-Nemrat, S. Venkatraman, Deep learning approach for intelligent intrusion detection system, IEEE Access 7 (2019) 41525–41550.
[34] T.-T.-H. Le, Y. Kim, H. Kim, Network intrusion detection based on novel feature selection model and various recurrent neural networks, Appl. Sci. 9 (2019) 7:1392 1–29.
[35] N. Shone, N.N. Tran, D.P. Vu, Q. Shi, A deep learning approach to network intrusion detection, IEEE Trans. Emerg. Top. Comput. Intell. 2 (2018) 1, 41–50.
[36] B. Yan, G. Han, Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system, IEEE Access 6 (2018) 41238–41248.
[37] S. Kalsi, H. Kaur, V. Chang, DNA Cryptography and deep learning using genetic algorithm with nw algorithm for key generation, J. Med. Syst. 42 (2018) 1–17.
[38] W. Bul'ajoul, A. James, S. Shaikh, A new architecture for network intrusion detection and prevention, IEEE Access 7 (2019) 18558–18573.
[39] X. Zhang, J. Chen, Y. Zhou, L. Han, J. Lin, A multiple-layer representation learning model for network-based attack detection, IEEE Access 7 (2019) 91992–92008.
[40] S. Anwar, J.M. Zain, M.F. Zolkipli, Z. Inayat, S. Khan, B. Anthony, V. Chang, From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions, Algorithms 10 (2017) 1–24.
[41] E. Kabir, J. Hu, H. Wang, G. Zhuo, A novel statistical technique for intrusion detection systems, Future Gener. Comput. Syst. 79 (2018) 303–318, part 1.
[42] X. Gao, C. Shan, C. Hu, Z. Niu, Z. Liu, An adaptive ensemble machine learning model for intrusion detection, IEEE Access (2019) 82512–82521.
[43] A. Binbusayyis, T. Vaiyapuri, Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach, IEEE Access 7 (2019) 106495–106513.
[44] Y. Zhang, P. Li, X. Wang, Intrusion detection for IoT based on improved genetic algorithm and deep belief network, IEEE Access 7 (2019) 31711–31722.
[45] A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for Internet of Things, Future Gener. Comput. Syst. 82 (2018) 761–768.
[46] Z. Chiba, N. Abghour, K. Moussaid, A. El omri, M. Rida, Intelligent approach to build a deep neural network based IDS for cloud environment using combination of machine learning algorithms, Comput. Secur. 86 (2019) 291–317.
[47] Y. Yang, K. Zheng, C. Wu, Y. Yang, Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network, Sensors 19 (11) (2019) 2528, http://dx.doi.org/10.3390/s19112528, 1–20.
[48] V. Chang, M. Ramachandran, Towards achieving data security with the cloud computing adoption framework, IEEE Trans. Serv. Comput. 9 (2016) 138–151.
[49] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer-Verlag New York, NY, USA, 2001.
[50] S. Revathi, A. Malathi, A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection, Int. J. Eng. Res. Technol. 2 (2013) 1848–1853.

Minh Tuan Nguyen received the B.S. degree from the Post & Telecommunications Institute of Technology, Hanoi, Vietnam, in 2004, the M.S. degree from Hanoi University of Science and Technology, Hanoi, Vietnam, in 2008, both in electronics and telecommunications engineering, and the Ph.D. degree at the Gwangju Institute of Science and Technology, Gwangju, South Korea, in 2018. His research interests include wireless communication systems, network security, biomedical signal processing, machine learning, deep learning, optimization, and biomedical application design.

Kiseon Kim received the B.Eng. and M.Eng. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1978 and 1980, respectively, and the Ph.D. degree in electrical engineering systems from the University of Southern California, Los Angeles, CA, USA, in 1987. From 1988 to 1991, he was with Schlumberger, Houston, TX, USA. From 1991 to 1994, he was with the Superconducting Super Collider Lab, TX. In 1994, he joined Gwangju Institute of Science and Technology, Gwangju, Korea, where he is currently a Professor. He is the member of The National Academy of Engineering of Korea, the Fellow IET and the Senior Editor of IEEE Sensors Journal. His current interests include wideband digital communications system design, sensor network design, and analysis and implementation both at the physical layer and at the resource management layer, biomedical applications design.