



Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection

Md Arafatur Rahman¹ · A. Taufiq Asyhari² · Ong Wei Wen¹ · Husnul Ajra³ · Yussuf Ahmed² · Farhat Anwar⁴

Received: 14 September 2020 / Revised: 8 December 2020 / Accepted: 13 January 2021 /
Published online: 8 March 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

The rapid advancement of technologies has enabled businesses to carry out their activities seamlessly and revolutionised communications across the globe. There is a significant growth in the amount and complexity of Internet of Things devices that are deployed in a wider range of environments. These devices mostly communicate through Wi-Fi networks and particularly in smart environments. Besides the benefits, these devices also introduce security challenges. In this paper, we investigate and leverage effective feature selection techniques to improve intrusion detection using machine learning methods. The proposed approach is based on a centralised intrusion detection system, which uses the deep feature abstraction, feature selection and classification to train the model for detecting the malicious and anomalous actions in the traffic. The deep feature abstraction uses deep learning techniques of artificial neural network in the form of unsupervised autoencoder to construct more features for the traffic. Based on the availability of cumulative features, the system then employs a variety of wrapper-based feature selection techniques ranging from SVM and decision tree to Naive Bayes for selecting high-ranked features, which are then combined and fed into an artificial neural network classifier for distinguishing attack and normal behaviors. The experimental results reveal the effectiveness of the proposed method on Aegean Wi-Fi Intrusion Dataset, which achieves high detection accuracy of up to 99.95%, relatively competitive to the existing machine learning works for the same dataset.

Keywords Attack classification · Centralized intrusion detection · Deep learning · Feature selection · Impersonation attack · Internet of things · Wi-Fi

✉ Md Arafatur Rahman
arafatur@ump.edu.my; arafatur.rahman@ieee.org

Extended author information available on the last page of the article.

1 Introduction

The rapid advancement in information and communication technologies has provided many advantages to system users but these technologies have many vulnerabilities that could be exploited by network intruders. Recent trends include the deployments of Internet of Things (IoT) [31] and other smart devices such as mobile phones, which have given attackers increased opportunities to exploit the vulnerabilities on this devices and to ultimately compromise the underlying network. To make a better use of these IoT devices, many organizations have deployed Wi-Fi networks, which also contribute to added security challenges including the possibility of fake access points and other emerging threats. These security challenges necessitates the need to develop a strong intrusion detection system (IDS) [13] capable of detecting various attacks.

IDS corresponds to a typical programmable computer application, which monitors all activities in the network and identifies unauthorized accesses [25] as well as malicious actions as intrusions that compromise with the confidentiality, availability and integrity of the system. Unauthorized accesses and malicious attacks are increasing day by day along with the enhancement of IoT applications as part of the network data exchanges [17]. Due to the lack of balance between the network vulnerabilities and the progression of security mechanism for IoT applications [8], conventional intrusion detection processes have been seen to be not as fully reliable as possible. This presents challenges since network data security is considered as an important factor for modern information communication, and data storage to drive real-world services within the society.

Recent research has demonstrated the advance in IDS research, including the enforcement of IDS [37] to observe the whole network security in real time. Statistically, a pre-dominant direction of IDS work is to design a system that improves the accuracy of detection and reduces the rate of false alarms. An IDS often analyzes the consequences after observing the facts to identify the mistrustful events and send the report for further decision making of detecting intrusions or suspicious activities in the IoT network [12]. Emerging research interests have also shown the proposal of various machine learning approaches, such as fuzzy logic, neural networks, and support vector machines, [11] to design IDS in response to detect the growing cyber-attacks [14] against secured IoT data communication links.

This work considers utilization of machine learning to address the issues of impersonation attacks within the Wi-Fi-based IoT networks. Building upon recent works on the AWID dataset that is well representative for such attacks (see, e.g., [3, 22, 23, 32]), we aim to design a proper machine learning-enabled mechanism to improve the performance of these existing works. More specifically, our contributions include the following:

1. We propose a machine learning-enabled model that offers combination of effective wrapper-based feature selection techniques for an intelligent IDS.
2. The model will first utilize feature extraction to generated additional features to investigate more characteristics from possible impersonation attacks.
3. The model will then select important attributes from multiple wrapper-based algorithms, namely Support Vector Machine (SVM), detection tree C4.5 and Naive Bayes (NB), for effective feature selection. The top-ranked attributes across different techniques will then be sequentially combined to improve feature list for classification based on the individual algorithm performance.
4. We finally use the classifier in the form of Artificial Neural Network (ANN) to distinguish impersonation attacks from the normal data traffic.

SVM is a form of supervised machine learning that can be used for wrapper-based feature selection. In the context of security studies, SVM has been shown for detecting intrusion attacks on the network traffic [41]. The optimization of the feature weights and SVM parameters [30, 38] are performed to select the subset of optimal features. The decision tree C4.5 classifier [26] is one of the tree models that can be used for IDS training and inherently provide a feature selection function. C4.5 uses the default information gain ratio to select the best splitting attributes in order to handle missing values and continuous data [33]. The NB technique [15] can be used to detect feature attributes, which will separate attack traffic over normal traffic and it has been shown to be effective in many aspects. The classifier training of NB calculates the occurrence frequency within the training samples and classifies the feature items by the classifier. The NB based model was suggested for better detection accuracy in the Network security attacks [4].

This paper specifically combines different top-ranked features identified by these three algorithms individually. The proposed feature combining method then utilizes the combined top-ranked features to train an artificial neural network (ANN)-based classifier. To validate the method and model, we leverage numerical experiments using standard machine learning metrics as the performance objectives. Using the widely-used AWID dataset [22], our experiments demonstrate the effectiveness of the feature combining method to achieve a high level of accuracy for accurately detecting impersonation attacks and distinguish them from the normal network traffic.

The remaining sections are structured as follows. Section 2 discusses the existing works related to machine learning-based IDS and shows the comparison of previous studies. The overall description of the proposed methodology is provided in Section 3. This is then followed by Section 4 where numerical experiments are used to examine the proposed method using standard classification performance metrics. Section 5 provides a summary of the paper with an overview of future works.

2 Related work

A number of previous authors have discussed features selection techniques, classification and machine learning algorithms. Aminanto et al. [3] built a novel Deep-Feature model based on extraction and selection techniques to combine the extracting stacked feature with weighted feature selection using well-referenced benchmark dataset such as AWID [22] including C4.5, ANN and SVM architectures based D-FES and Dataset Pre-processing function. The data processing function consists of dataset normalization and balancing. In [45], the authors trained the target values to visualize the learning effect and to realize the pre-training effect for different hidden layers of stacked auto-encoder as feature extraction layer using back-propagation on MNIST database. and the convergence speed is increased for faster learning process. In addition, the classification accuracy results is different due to small training error in the fine-tuning process. Android based malapp detection system is performed in [42] which explore the permission-reduced risks on systematic three levels. In addition, the authors ranked the risk of all the individual permissions and a group of collaborative permissions using T-test, mutual information and Correlation Coefficient as well as analyzed the risk permissions subsets with Sequential Forward Selection and Principal Component Analysis. Also random forest, SVM and decision trees were conducted to construct the determination of malapps with risky permissions. Lastly they analyzed

the effectiveness in depth and the limitations on malapps detector using only permission requests.

Javadpour et al. [18] proposed feature selection and intrusion detection system with the linear correlation and mutual information which the authors claimed to have increased the exactitude of the intrusion determination rate in cloud computing using KDD99 database for automatic detection of new and suspicious patterns based decision tree, random forest and CART algorithms with neural network. In [35], the authors proposed multi-feature collaborative model with decision fusion to analyze multiple features of benign and malware families, and to design ensembles of classifier based mechanism for better scalable detection as well as to present collaborative decisions by comparing state-of-the-art ensemble learning techniques. Firdaus et al. [10] used the static exploration including genetic search (GS) algorithm, five machine learning architectures such as random forest, Naïve Bayes, functional trees (FT), multilayer perceptron (MLP) and J48 for feature selection among 106 strings in android malware detection system where FT provide utmost accuracy and high true positive rate than other classifiers.

Parker et al. [29] constructed DEMISe-RBFC model and DETEReD model using deep-structured stacked autoencoder and rank based mutual information theoretic feature selection that provide better accuracy than the state-of-the-art architectures of machine-learning with lesser computational expense and perform upper levels of interpretability and transparency to test the best-referenced intrusion detection dataset against AWID dataset for IoT intrusion detection. The authors [34] analyzed three approaches namely correlation feature selection, gain of information and gain ratio and to assemble best peak features selection associated with ranking features for high detection rate. Also, the KNN, NB and Neural Network basis Multilayer Perceptron were used to select six features after testing 41 features using KDD dataset for classification and K-Nearest Neighbor acquires high accuracy in intrusion detection system. In [43], a semi-supervised SVM mentioned for classifying the driving behavior and variances from different amounts of labeled data using a k-means clustering method and quasi-Newton algorithms which provides high classification accuracy and reduces the labeling effort among huge amounts of unlabeled data.

A distribution preserving kernel SVM (DiP-SVM) was proposed by Singh et al. [36] that acquires comparable classification performance than sequential SVM on several benchmark datasets in local decision boundaries and global decision boundaries as well as mitigates communication overhead between partitions for faster execution in cloud environments on large datasets. Al-Jarrah et al. [2] the authors constructed a randomized data partitioning model RDPLM for massive volume networks to remove redundant and irrelevant features using voronoi-clustering and feature ranking based data separating algorithms on benchmark botnet dataset. Also, the authors evaluated and compared RDPLM with multilayered deep neural network sequential minimal optimization (SMO), DNN, randomTree (RTree), REPTree, C4.5 and decision tree for botnet intrusion detection. Additionally, they achieved high detection accuracy and reduced the model computational cost. Ouyang et al. [28] constructed a unified deep model (UDM) for pedestrian identification. [5] proposed DBN based deep learning approach for feature extraction and image classification using indian pines and pavia datasets.

Kasun et al. [21] explored Extreme Learning Auto-Encoder using USPS dataset to recognize handwritten digits, NORB and CIFAR-10 object identification datasets in dimension reduction structure. Alamiedy et al. [1] provided an anomaly-based IDS associated with grey wolf optimization algorithm. In their work they used SVM to determine the competence of the selected features resulting in a large scale classification exactness with

appeasement features subset for different attacks compared with other existing approaches. Jaber and Rehman [16] demonstrated FCM–SVM based IDS to connect fuzzy c-means-clustering procedure and SVM using NSL-KDD dataset in cloud computing environment which provides high detection accuracy and low false alarm rates based on the performance metrics. In this paper, we have performed the comprehensive observations of available literature on feature selection techniques.

Chiew et al. [6] analyzed the ensemble feature selections among 48 features which are extracted from URLs and HTML for phishing detection. The authors collected 5000 phishing and 5000 legitimate webpages for their dataset based on URL and HTML documents. They designed an ensemble feature selection based hybrid framework (HEFS) to utilize the function gradient algorithm of cumulative distribution with obtaining a baseline feature set for phishing detection using machine learning. Verma and Ranga [39] assessed the performance of ML classifiers using datasets NSL-KDD, UNSWNB15 and CIDDSS-001 to detect the assaults against IoT and measured the performances of statistical analysis among classifiers. Li et al. [24] developed a system of intrusion detection in industrial IoT using multi-CNN fusion process on NSL-KDD dataset to apply binary and multiclass classification as deep learning procedures.

A list of these works and the categories they addressed can be found in Table 1. Herein, it has been mentioned their purposes and applying techniques as well as datasets. In this table, it is made to list the benefits and limitations of their work in order to understand the results of previous work.

3 Proposed method for feature combining

Feature learning or featured selection is a technique which ranks according to their relevance. Feature selection methods leads can increase the capability of an IDS via appropriate metrics improvement. Features selection may be categorized into two groups, namely wrapper-based and filter-based methods. Filter based feature selection are simple, less complex and filters the best attributes depending on their optimal influence [34]. Only the most important attributes are selected attributes and classified later to reduce complexity and improve the accuracy of the model. However, wrapper-based feature learning typically employs an algorithm to search for the most important features. Herein various subsets of features are assessed using the final classification criterion with the best subset being ultimately chosen [20]. As overall, the objective of features selection is to select the most influential attribute to target with a lower dimension which use of classification step. Moreover, the function of feature extraction is to compress the existing features and reconstruct the similar features. The benefit of features extraction is to increase efficiency of the classifier.

We implement a deep feature extraction and feature selection combination technique. Figure 1 depicts the overall architecture of features combining and step-by-step processes of the technique for binary classes, namely normal traffic and impersonation attacks. Pre-processing is a necessary step and is performed at earliest phase, which includes normalization and balancing dataset stages. The procedure for this phase will be explained in the section below in details. We adopt both feature extraction and combination of best ranking selection techniques in IDS. We apply both feature extraction and combination of top of the ranking selection process in IDS. We consider feature extraction using Sparse/Stacked Auto encoder (SAE) structure [45], incorporating subsequent pairwise hidden layers to maximize

Table 1 Comparative study of previous work

References	Purposes	System/Techniques	Dataset	Benefits	Limitations
[3]	impersonation attacks detection	D-FES	AWID	most accurate detection of impersonation attacks	i) emphasized only impersonation attack but not others ii) use inadequate computing power, memory, and power supply
[45]	To achieve high learning capability	Stacked autoencoder	MNIST	i) increase the convergence speed ii) faster learning process	Difficult to train the process of neural network for increasing hidden layers
[42]	Detect the permission-reduced risks from malapps	Android based malapp detection system	Benign apps and malapps	identify risky permission in android applications	Detect malicious using only permission requests
[18]	increase the accuracy of intrusion detection	Feature selection and intrusion detection system	KDD99	Intrusion Detection in Cloud Environment	Only perform in Cloud Environment
[35]	Identify Malware by extracting multiple sets of features	MCDF	benign and malware families	Malware detection on Android devices	Not recognize best set of features.
[10]	evaluate the optimal features	Genetic search(GS) based method	benign and malware	Malware detection on Android platform	Not conduct dynamic analysis
[29]	IoT impersonation intrusion detection	DEMISE-RBFC and DETEReD	CLS-AWID	Provide lower computational cost, interpretability and transparency	emphasized only impersonation attack but not others
[34]	Combining best features selection	correlation feature selection, gain of information, gain ratio based IDS	KDD dataset	Evaluate the intrusion detection rate and accuracy	Time consuming for training classifier
[43]	Classifying the driving behavior and variances	semi-supervised support vector machine (S3VM)	Driving dataset	require a very low labeling effort	Working on a curvy road but not on weather or traffic

Table 1 (continued)

References	Purposes	System/Techniques	Dataset	Benefits	Limitations
[36]	Acquire comparable classification performance	distribution preserving kernel support vector machine (DiP-SVM)	Several benchmark datasets	mitigate communication overhead between partitions in cloud environments on large datasets	Not recognize best benchmark datasets for DiP-SVM
[44]	hierarchical semantic features learning automatically from raw remote sensing (RS) data	SDSAE and FV based deep filter banks	RSSCN7 and UC Merced data sets	Used in the field of remote sensing (RS) image processing	reduce the training complexity for linear SVM but not for nonlinear classifier
[2]	eliminate redundant and irrelevant features	RDPLM	benchmark botnet dataset	High intrusion detection and minimize model computational expense	Low computing power and memory.
[28]	maximize the strength of framework components	unified deep model (UDM)	Caltech dataset and ETH dataset	Pedestrian detection	Not for larger-scale training sets
[5]	Extract and classify the deep and invariant features for 3-D hyperspectral image	deep belief network (DBN)	Indian Pines and Pavia	huge potential for hyperspectral data classification	limited training samples
[21]	Reduce the noise or irrelevant information of linear and nonlinear dimension	ELM-AE and SELM-AE	USPS, CIFAR-10, NORB	Lower computational complexity	Use single hidden layer
[1]	identify the relevant feature and abnormal behavior	grey wolf optimization (GWO)	NSL-KDD	High classification accuracy	Calculate only classification accuracy but not analysis detection rate and classification error
[16]	Increase the detection accuracy	FCM-SVM based IDS	NSL-KDD	high detection accuracy and low false alarm rates	Only using NSL-KDD dataset
[19]	Increase the accuracy over other methods	FFDNN-IDS	NSL-KDD	Perform the multiclass and binary classification.	Only using NSL-KDD dataset

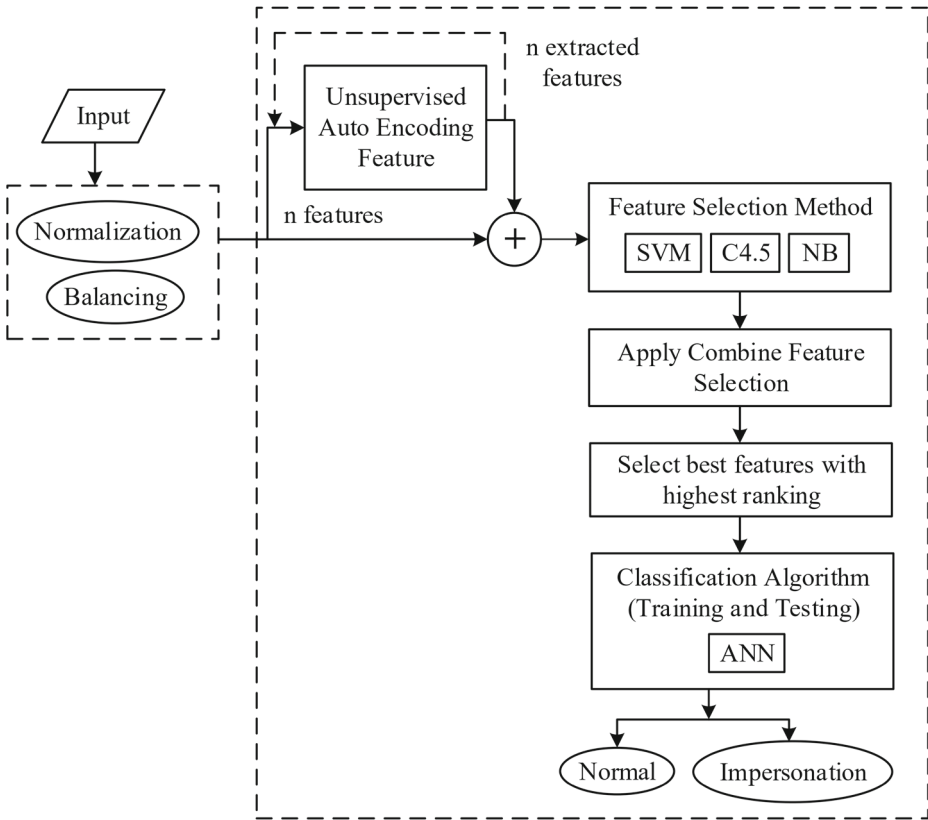


Fig. 1 Architecture of combining best features for centralized IDS with binary classification, namely normal traffic and impersonation attacks

the capability of learning and improve the running time. Then, the outputs of SAE with 50 extracted features are joined together with the existing features of the AWID dataset [3]. Moreover, appropriate combination of feature selection techniques are employed by C4.5, SVM, and ANN algorithms that include well-referenced machine learners for the design of candidate models.

3.1 Feature extraction

Sparse or Stacked Auto Encoder (SAE) corresponds to an unsupervised learning technique for building new features using non labelled data. An Auto Encoder (AE) itself is a type of a neural network that has connections with other neurons in the hidden layer. Figure 2 shows an AE with 154 features from original inputs and 50 neurons in encoder layers which extracts new features using auto encoder and decoder by inputting the data into the neurons through hidden layer, and every neuron within the encoder layer are often calculated from input and compressed into code and extract important details in it. The AE will compress the original features into code, and captured relevant and detailed data by compressing them into new similar features.

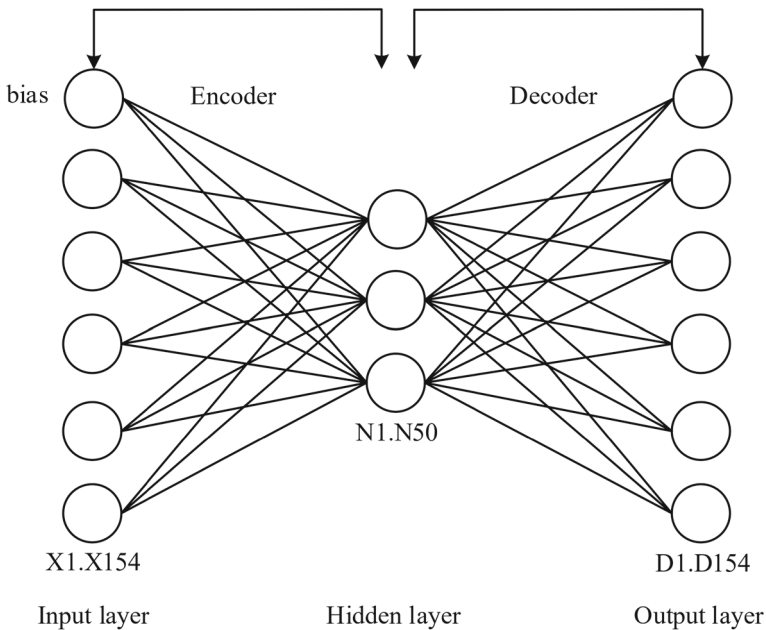


Fig. 2 AE network illustration with input, hidden and output layers. The construction herein is symmetric

An AE or a neuron of an ANN is a computerized analysis unit, wherein the input layer accepts parameters x_1, x_2, \dots, x_n and the output layer yields the variables O_1 and O_2 as a hypothesis denoted as function $h_{W,b}(x)$. Herein the subscript W is called weight, whilst b denotes the bias. The learning algorithm adjusts to suit the training data in the AE or neural network. After that, with the extracted data from compression in the hidden layer, decoder layer will reconstruct new features with similar, close result as input data.

Formally, the AE can be represented using a function that maps vectors from input x into z . This can be captured using the following equation

$$z = h^{(encoder)}(W^{(encoder)}x + b^{(encoder)}). \tag{1}$$

Similarly, towards the end of the processing, the decoder aims to estimate the input x using z and this can be precisely written as:

$$x = h^{(decoder)}(W^{(decoder)}z + b^{(decoder)}) \tag{2}$$

The AE parameters can be predicted using back propagation algorithm where from an outside perspective the hypothesis $h_{W,b}(x)$ for training is specified to x . This process necessitates the AE to learn a feature that meets the objective and constraint. The AE algorithm takes the assumption that achieving local optimized weights and biases at every neuron which provides the overall optimal output by gaining knowledge of a compressed representation of the given scope. To improve the accuracy of the feature, we have used SA for feature extraction. SAE uses the original input to learn the first AE and build the second AE by using first encoder which is shown in Fig. 3. SAE architecture enables progressive refinement of the quantity of hidden neurons in the encoder layer to reduce the greediness problem of the algorithm.

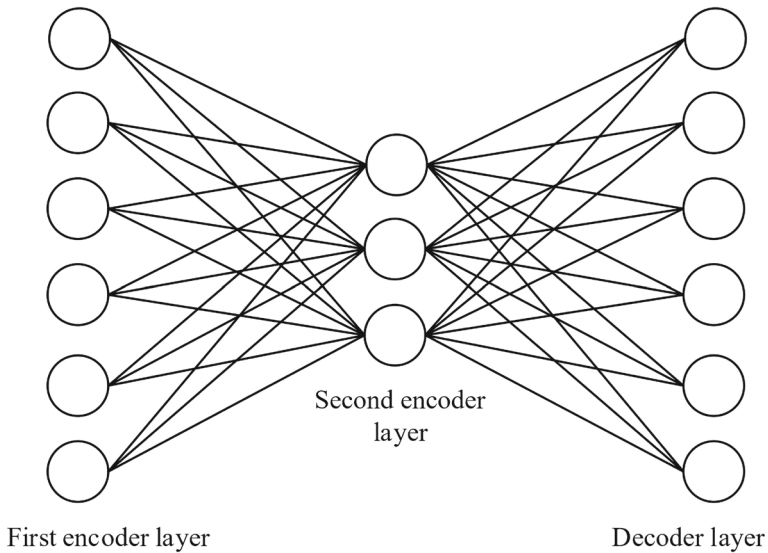


Fig. 3 Stacked Autoencoder using first encoder layer to train second encoder layer

In Fig. 4, it is described an SAE with hidden layers for two target classes. This SAE structure aims to overcome over-fitting problem. For SAE, a sparsity penalty function often utilizes Kullback-Leibler (KL) divergence [7], which can be written as [3]:

$$\Omega_{sparsity} = \sum_{i=1}^h D_{KL}(P || \hat{P}_i) \tag{3}$$

$$= \sum_{i=1}^h \left[P \log \left(\frac{P}{\hat{P}_i} \right) + (1 - P) \log \left(\frac{1 - P}{1 - \hat{P}_i} \right) \right]. \tag{4}$$

Where, D_{KL} is the function of Kullback-Leibler (KL) divergence, when the desired value is denoted by P and activation value of average output is \hat{P}_i with number of hidden layer neurons h .

For the feature extraction part prior to feature selection and combining, we specifically follow the Deep-Feature Extraction and Selection (D-FES) model in [3] where an SAE architecture of 154:100:50 was used to analyze the reduced CLS version of the AWID dataset [22]. This means that we use 100-neuron to train the encoder from the existing 154 features. On the other hand, for the second layer, we use 50-neuron and train from the first 100-neuron encoder.

3.2 Feature selection

From Fig. 1 that demonstrates the architecture of our proposed method, it is shown three different feature selection algorithms, namely SVM, Decision Tree C4.5 and Naïve Bayes. These algorithms are expected to select the different ranking of important subset of the features. The details of each feature selection techniques are represented below.

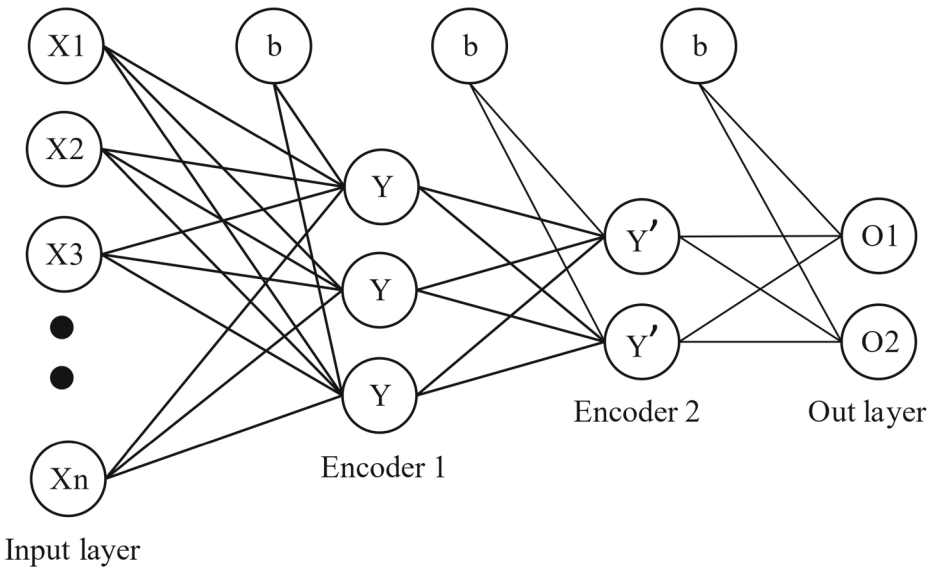


Fig. 4 An illustration of SAE network. Here we have two hidden layers for two output labels

1. SVM belongs to supervised learning and corresponds to a classification method for both linear and non linear data. It can be applied to perform classification and regression. Suppose we have N number of input features from two targets, the SVM will plot the features data in the graph using the value of predictor on the two dimensional axis or n -dimensional space. The basic concept of SVM classification process is to find the hyperplane that segregate binary classes [40]. Even though the SVM has the ability to deal with non linear classifications that are used N -dimensional hyperplane to distinguish classes. In this work, we choose normal SVM to perform the classification because it is enough to segregate our attribute. The key computational properties of SVM include data instances that are nearest to the decision boundary and that are supportive vectors.
2. Decision Tree C4.5 has properties of being resilience to noisy observations and capable to capture disjunctive expressions. This algorithm forms a k -array tree structure by counting the value of gains from input data by each node, where the biggest gain is to be used as an initial node or the root node [26, 33]. For tree construction, C4.5 invokes a greedy algorithm following a recursive divide-and-conquer method [26, 33].
3. Naïve Bayes (NB) represents a possible type of probabilistic supervised classification method based on the Bayes theorem [15, 27]. The equation that governs Bayes theorem can be expressed as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (5)$$

Herein A can denote the event that is happening, and we need to find out its probability, while B is the event that has already happened before. $P(A)$ represents the overall probability of occurrence for A before seeing any proof. The attribute value of

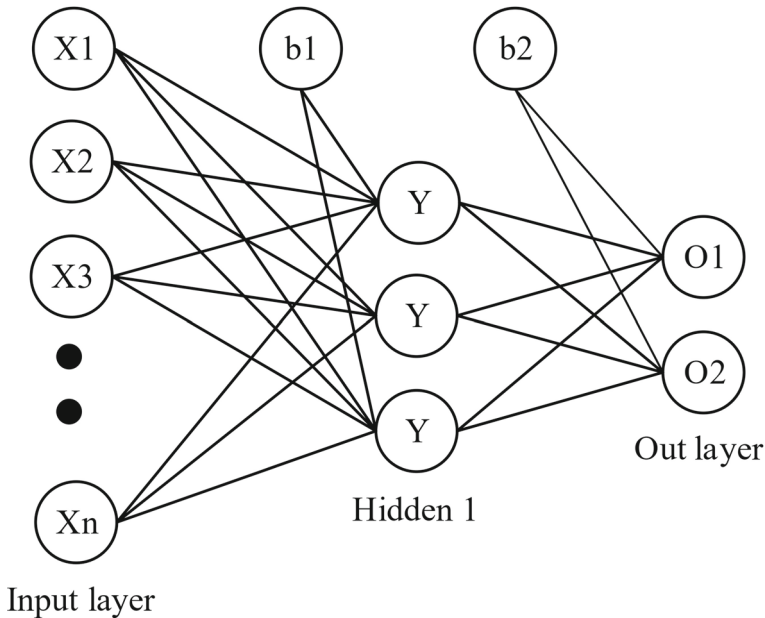


Fig. 5 The construction of ANN for classification

an unknown instance of B acts as the proof. On the other side, $P(A|B)$ depicts the posterior probability of A after seeing the proof in B

One of the principles in Bayes theorem is every pair of the features is classified independently between each other. The basic idea of this technique is to find the likelihood of an event taking place using the likelihood of another event that has already happened. Bayesian classification possesses capability that is close to the decision tree and neural network in terms of performance. Bayesian classification is shown to achieve accuracy and speed in applying large data scenario.

3.3 Classification

The ultimate classification algorithm employs an Artificial Neural Network (ANN), which uses multilayer feed-forward network to correlate with the neurons in the hidden layer [9]. The ANN is trained with normal and impersonation attacks only. The construction of this ANN is shown in Fig. 5 which consists of input-output layers as well as only one hidden layer in the middle. Herein b_1 and b_2 denote independent weight values to help the input data for fitting the model.

The main responsibility of activation function such as Sigmoid or ReLu is to determine the exact output of ANN. The neuron with input features data are related to activation function. The function will add information and perform activation from different sources into different element from basic input weights. The function of non-linear activation function is to ensure the neuron which back-propagate is always bring the actual output of neuron which is result of stimulation of neurons and can be control.

4 Numerical evaluation and discussion

In this section, we discuss the numerical experiments of the studied feature combining method that has been conducted to evaluate the performance including accuracy, true positive and other factors. In this experiment, we have chosen the AWID Dataset [22] as a proper dataset which is constituted from Wi-Fi traffic data collection. In this experiment, the tools are used to evaluate the methodology are MATLAB R2018b, AWID dataset, Weka 3.9 and the hardware uses Intel R Core I5-6200U 2.3 GHz with 4GB dedicated RAM laptop.

4.1 Data-preprocessing

The pre-processing part is shown at the first phase of Fig. 1. This part processes the dataset before we proceed to the next phase. The AWID dataset [22] contains a group of big dataset such as AWID-CLS-R-Trn, AWID-CLS-R-Tst, AWID-CLS-F-Trn, AWID-CLS-F-Tst. In this experiment, we have chosen AWID-CLS-R-Trn dataset where “R” represent a reduced dataset. The AWID also comes with AWID-ATK-R-Trn set, wherein the “ATK” contains 16 target classes. The 16 type of attack target classes are contained in “ATK” dataset with more details like Evil Twins, Honeypot, EvilTwins and other attacks. The “CLS” data have the compressed version of attack categories which are injection, flooding, impersonation attack and normal class which are listed in “ATK” dataset. Based on the size, the “CLS” is enough to perform the simulation and evaluation to conduct research. The reduced “CLS” dataset provide more simplicity for us to proceed all phase. The “CLS” dataset contain 1.7 million rows with 154 column attributes and 1 column target class. After selecting the type of dataset, since we only test for impersonation attack, thus we will remove other attack classes in the dataset which are flooding and injection attack target. After removing other classes, we only left normal and impersonation classes. In left portion of the data, we have chosen 97044 rows data in it and with ratio 50:50 impersonation attack class and normal as target class.

We next need to process data normalization. Hence, every attribute of data has equal range of values. To avoid an excessive amount of influence of various scales, we use a mean range method to linearly normalize the data between 0 and 1, i.e.,

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}. \quad (6)$$

Here x_i and z_i provide the current input and the result of the normalization, respectively whilst $\min(x)$ and $\max(x)$ represents the minimum and maximum of the attribute, respectively. After normalization, we finally split the dataset into 70% being the training set and 30% being the testing set.

4.2 Test metrics

We validate the effectiveness of our proposed feature combining method using standard machine learning classification metrics [3, 32], namely Accuracy (Acc), Detection Rate (DR), False Alarm Rate (FAR), Mcc, and CPU time to build model (TBM). Herein Acc measures the ability to correctly classify the labels (normal traffic or attacks). The DR measures the number of actual attacks in comparison to the total attacks being detected. False Alarm Rate (FAR) corresponds to the number of normal traffic instances that are detected as other than normal instances.

F1 score is demonstrate to compute the harmonic mean of Recall and Precision to present a model accuracy measure. MCC score means matthews correlation coefficient which is applied for the quality measurement of binary classifications in the area of machine learning Mcc indicates the correlation measure of data from detection and observation [2]. A list of equations that capture these various performance metrics are given in the following as used in [3].

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$DR = \frac{TP}{TP + FN} \quad (8)$$

$$FAR = \frac{FP}{TN + FP} \quad (9)$$

$$Mcc = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \quad (10)$$

In the above equations, we have denoted the following.

- TP is the quantity of attack instances accurately classified in the same class.
- TN is the quantity of normal instances accurately classified in the same class.
- FN is the quantity of attack instances incorrectly classified as normal class.
- FP is the quantity of normal instances incorrectly classified as attack.

4.3 Feature selection

The previous D-FES research has shown the effectiveness of feature selection using a wrapper-based selection technique. Since the wrapper-based technique can select effective attribute in dataset, we propose a method to use a combination of three algorithms to select features. Herein the considered features will be based on widely-used learning algorithms in previous works, namely SVM, C4.5, and NB. In this paper, we will use wrapper-based techniques for SVM, C4.5, and NB, respectively, and generate ranking of the 204 features, and we will choose 10% of the attributes, accounting for around 20 features to aid in the classification process. The first two wrapper-based techniques will choose top 7 ranked features, whilst the last technique will only choose 6 features.

One of the problems is that these algorithms might select the same features. In order to avoid this, it has been applied the following aspects to perform the combination of ranked features.

- Firstly, it has been chosen the top 7 ranked features from SVM.
- This is then followed by C4.5 that chooses top 7 ranked attributes; if an attribute selected clashes with one of the SVM top 7 selected attributes, it has been skipped this and moved to the next attribute within the list.
- The same process is applied to the NB, i.e., it skips the repeated features selected previously by SVM and C4.5 and identifies the complementary 6 top ranked features as appropriate.

Table 2 summarizes the features selected by a variety of wrapper-based selection techniques.

As taken from the AWID dataset used in [22], Table 3 the feature names with the descriptions. As underlined in [23], timing features within the dataset are not very useful in designing IDS since it is less reasonable to identify the timing of attacks in realistic scenarios.

Table 2 Features selected using various wrapper-based algorithms

Feature selection method	Rank search
SVM	47, 107, 67, 112, 108, 82, 94, 78, 141, 122, 70, 111, 7, 98, 73, 79, 130, 90,4, 129
C4.5	38, 79, 8, 67, 82, 107, 76, 4, 1, 2, 3, 5,6, 7, 9, 10, 11, 12, 13, 14
NB	8, 4, 50, 73, 70, 94, 1, 2, 3, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
Combination top rank	47, 107, 67, 112, 108, 82, 94, 38, 79, 8,76, 1, 2, 3, 50, 73, 70, 10, 11, 12

Therefore, we do not choose features 4, 5, 6, 7 since they represents “frame.time_epoch”, “frame.time_delta”, “frame.time_delta_displayed” and “frame.time_relative”, respectively, as listed in the Wireshark reference document.

Table 3 Identified feature set with feature names and descriptions as taken from the AWID dataset used in [22]

Index	Feature name	Description
1	frame.interface_id	Interface id
2	frame.dlt	WTAP.ENCAP
3	frame.offset_shift	Time shift for this packet
4	frame.time_epoch	Epoch Time
5	frame.time_delta	Time delta from previous captured frame
6	frame.time_delta_displayed	Time delta from previous displayed frame
7	frame.time_relative	Time since reference or first frame
8	frame.len	Frame length on the wire
10	frame.marked	Frame is marked
11	frame.ignored	Frame is ignored
12	radiotap.version	Version of radiotap header format
38	radiotap.mactime	Value in microseconds of the MAC's Time Synchronization Function timer when the first bit of the MPDU arrived at the MAC
47	radiotap.datarate	Speed this frame was sent/received at
50	radiotap.channel.type.cck	Channel Type Complementary Code Keying (CCK) Modulation
67	wlan.fc.subtype	Subtype
70	wlan.fc.retry	Retry
73	wlan.fc.protected	Protected flag
76	wlan.ra	Receiver address
79	wlan.sa	Source address
82	wlan.seq	Sequence number
94	wlan_mgt.fixed.capabilities.preamble	Short preamble
107	wlan_mgt.fixed.timestamp	timestamp
108	wlan_mgt.fixed.beacon	Beason interval
112	wlan_mgt.fixed.auth_seq	Authentication SEQ

Table 4 Classification results using ANN

Selected features	DR (%)	FAR (%)	ACC (%)	F_1 (%)	Mcc (%)	TBM (s)
Combination	99.90	0.001	99.95	99.95	99.90	243.28
SVM top 20	99.93	0.001	99.89	99.89	99.77	198.41
C4.5 top 20	99.88	0.001	99.91	99.91	99.82	148.64
NB top 20	99.17	0.03	98.29	98.30	96.60	143.21

4.4 Classification

Table 4 presents the numerical experiment results for the ANN classifier using the chosen features in Table 2. By applying SVM on top 20 rank features, the numerical results has been acquired Acc (99.89%), F_1 (99.89%) and MCC (99.77%). For C4.5 technique, it has been achieved the numerical results Acc (99.91%), F_1 (99.91%) and MCC (99.82%). By performing NB technique, it has been attained the numerical results Acc (98.29%), F_1 (98.30%) and MCC (96.60%). But, the feature combining method herein achieves the highest Acc (**99.95%**), F_1 (**99.95%**) and MCC (**99.90%**). However, it needs the longest CPU time compared to other individual wrapper-based feature selection method to build the model. The top 20 features of NB has the fastest build model time with 143.21s but other performance is lower than SVM and C4.5 for top 20 features and Combination features. FAR is relatively low for the feature combining method, which aligns with those of SVM and C4.5. However, NB achieves a higher FAR (0.03) compared to the others but the figure still looks reasonable.

5 Conclusion

In this paper we have studied a new approach of designing machine learning-based IDS by employing a variety of feature selection techniques and effectively combining their selected features for accurate cyber attack classification. To ensure sufficient diversity in the pool of features, an SAE has been utilized to reconstruct more meaningful features that are significant for attack identification from the original features. This has been subsequently followed by invoking three algorithms, namely SVM, C4.5 and NB for individual wrapper-based feature selection in which proportional (almost uniform) outputs are combined to provide the 20 most effective features for classification. In the process of selecting these features, we have carefully examined similar selection and ensured optimal combining in the sense that it maximises the detection accuracy via disjoint selection. Herein the purpose is to eliminate irrelevant features and select the most influential features to train the machine learning model so as to boost the effectiveness of classification. By leveraging ANN for classification, the proposed method has been shown to outperform the existing algorithms in the context of standard metrics, namely ACC, F_1 and Mcc whose values are given by 99.95%, 99.95% and 99.90%, respectively. A shortcoming of this work is due to a longer time to build the model when compared to other individual algorithms using selected attributes. As this model is linked to the centralized approach with possible implementation in the cloud computers, such resource requirement would be less an issue due to a high processing power to overcome the model build time. For future work, our method has potential to be upgraded to

detect more various attacks beyond the impersonation attack considered in this work. Similarly, while our method can be deployed in IoT environments where low-resource sensing devices are connected to a powerful cloud, a challenge to address is when such a powerful machine is lacking. A relevant forward direction is to provision sophisticated collaborative computations among low-resource sensing, less powerful edge and powerful cloud devices.


Acknowledgements This paper is partially supported by the International Grants Number RDU192705 and UIC191516.

References

1. Alamiedy TA, Anbar M, Alqattan ZNM, Alzubi QM (2020) Anomaly-based intrusion detection system using multi-objective grey wolf optimisation algorithm. *J Amb Intell Hum Comput* 11(9):3735–3756
2. Al-Jarrah OY, Alhussein O, Yoo PD, Muhaidat S, Taha K, Kim K (2016) Data randomization and cluster-based partitioning for botnet intrusion detection. *IEEE Trans Cybern* 46(8):1796–1806
3. Aminanto ME, Choi R, Tanuwidjaja HC, Yoo PD, Kim K (2018) Deep abstraction and weighted feature selection for Wi-Fi impersonation detection. *IEEE Trans Inf Foren Sec* 13(3):621–636
4. Bhosale KS, Nenova M, Iliev G (2018) Modified naive bayes intrusion detection system (MNBIDS). In: 2018 international conference on computational techniques, electronics and mechanical systems (CTEMS). IEEE, pp 291–296
5. Chen Y, Zhao X, Jia X (2015) Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE J Sel Topics Appl Earth Observ Remote Sens* 8(6):2381–2392
6. Chiew KL, Tan CL, Wong K, Yong KS, Tiong WK (2019) A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Inf Sci* 484:153–166
7. Cover TM, Thomas JA (2006) Elements of information theory, 2nd edn. Wiley, Hoboken
8. Deng L, Li D, Yao X, Cox D, Wang H (2019) Mobile network intrusion detection for iot system based on transfer learning algorithm. *Clust Comput* 22(4):9889–9904
9. Elgammal MA, Mostafa H, Salama KN, Nader Mohieldin A (2019) A comparison of artificial neural network(ANN) and support vector machine(SVM) classifiers for neural seizure detection. In: 2019 IEEE 62nd international midwest symposium on circuits and systems (MWSCAS). IEEE, pp 646–649
10. Firdaus A, Anuar NB, Karim A, Razak MFA (2018) Discovering optimal features using static analysis and a genetic search based method for Android malware detection. *Front Inf Technol Elec Eng* 19(6):712–736
11. Gao X, Shan C, Hu C, Niu Z, Liu Z (2019) An adaptive ensemble machine learning model for intrusion detection. *IEEE Access* 7:82512–82521
12. Gassais R, Ezzati-Jivan N, Fernandez JM, Aloise D, Dagenais MR (2020) Multi-level host-based intrusion detection system for internet of things. *J Cloud Comput* 9(1):1–16
13. Gul A, Adali E (2017) A feature selection algorithm for IDS. In: 2017 international conference on computer science and engineering (UBMK). IEEE, pp 816–820
14. Han S, Xie M, Chen HH, Ling Y (2014) Intrusion detection in cyber-physical systems: techniques and challenges. *IEEE Sys J* 8(4):1052–1062
15. He D, Liu X, Zheng J, Chan S, Zhu S, Min W, Guizani N (2020) A lightweight and intelligent intrusion detection system for integrated electronic systems. *IEEE Netw* 34(4):173–179
16. Jaber AN, Rehman SU (2020) FCM–SVM Based intrusion detection system for cloud computing environment. *Cluster Comput*
17. Jan SU, Ahmed S, Shakhov V, Koo I (2019) Toward a lightweight intrusion detection system for the internet of things. *IEEE Access* 7:42450–42471
18. Javadpour A, Kazemi Abharian S, Wang G (2017) Feature selection and intrusion detection in cloud environment based on machine learning algorithms. In: 2017 IEEE International symposium on parallel and distributed processing with applications and 2017 IEEE international conference on ubiquitous computing and communications (ISPA/IUCC). IEEE, pp 1417–1421
19. Kasongo SM, Sun Y (2019) A deep learning method with filter based feature engineering for wireless intrusion detection system. *IEEE Access* 7:38597–38607
20. Kasongo SM, Sun Y (2020) A deep learning method with wrapper based feature extraction for wireless intrusion detection system. *Comput Secur* 92:101752

21. Kasun LLC, Yang Y, Huang GB, Zhang Z (2016) Dimension reduction with extreme learning machine. *IEEE Trans Image Process* 25(8):3906–3918
22. Koliás C, Kambourakis G, Stavrou A, Gritzalis S (2016) Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. *IEEE Commun Surv Tutor* 18(1):184–208
23. Lee SJ, Yoo PD, Asyhari AT, Jhi Y, Chermak L, Yeun CY, Taha K (2020) Impact: impersonation attack detection via edge computing using deep autoencoder and feature abstraction. *IEEE Access* 8:65520–65529
24. Li Y, Xu Y, Liu Z, Hou H, Zheng Y, Xin Y, Zhao Y, Cui L (2020) Robust detection for network intrusion of industrial iot based on multi-cnn fusion. *Measurement* 154:107450
25. Mighan SN, Kahani M (2020) A novel scalable intrusion detection system based on deep learning. *Int J Inf Secur*: 1–17
26. Mu Y, Liu X, Yang Z, Liu X (2017) A parallel C4.5 decision tree algorithm based on MapReduce. *Concur Comput Prac Exp* 29(8):e4015
27. Omrani T, Dallali A, Rhaimi BC, Fattahi J (2017) Fusion of ANN and SVM classifiers for network attack detection. In: 2017 18th international conference on sciences and techniques of automatic control and computer engineering (STA). IEEE, pp 374–377
28. Ouyang W, Zhou H, Li H, Li Q, Yan J, Wang X (2018) Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection. *IEEE Trans Patt Anal Mach Intell* 40(8):1874–1887
29. Parker LR, Yoo PD, Asyhari TA, Chermak L, Jhi Y, Taha K (2019) DEMISE. In: Proceedings of the 14th international conference on availability, reliability and security - ARES '19. ACM Press, New York, pp 1–10
30. Phan AV, Nguyen ML, Bui LT (2017) Feature weighting and SVM parameters optimization based on genetic algorithms for classification problems. *Appl Intell* 46(2):455–469
31. Rahman MA, Asyhari AT (2019) The emergence of internet of things (iot): Connecting anything, anywhere. *Computers* 8(2):40
32. Rahman MA, Asyhari AT, Leong L, Satrya G, Hai Tao M, Zolkipli M (2020) Scalable machine learning-based intrusion detection system for iot-enabled smart cities. *Sustain Cities Soc* 61:102324
33. Sahani R, Shatabdinalini, Rout C, Chandrakanta Badajena J, Jena AK, Das H (2018) Classification of intrusion detection using data mining techniques: 753–764
34. Salih AA, Abdulrazaq MB (2019) Combining best features selection using three classifiers in intrusion detection system. In: 2019 International conference on advanced science and engineering (ICOASE). IEEE, pp 94–99
35. Sheen S, Anitha R, Natarajan V (2015) Android based malware detection using a multifeature collaborative decision fusion approach. *Neurocomputing* 151:905–912
36. Singh D, Roy D, Mohan CK (2017) Dip-SVM : distribution preserving kernel support vector machine for big data. *IEEE Trans Big Data* 3(1):79–90
37. Sultana N, Chilamkurti N, Peng W, Alhadad R (2019) Survey on sdn based network intrusion detection system using machine learning approaches. *Peer-to-Peer Netw Appl* 12(2):493–501
38. Tao P, Sun Z, Sun Z (2018) An improved intrusion detection algorithm based on GA and SVM. *IEEE Access* 6:13624–13631
39. Verma A, Ranga V (2020) Machine learning based intrusion detection systems for iot applications. *Wirel Pers Commun* 111(4):2287–2310
40. Wadkar M, Di Troia F, Stamp M (2020) Detecting malware evolution using support vector machines. *Expert Syst Appl* 143:113022
41. Wang W, Du X, Shan D, Qin R, Wang N (2020) Cloud intrusion detection method based on stacked contractive Auto-Encoder and support vector machine. *IEEE Trans Cloud Comput*: 1–1
42. Wang W, Wang X, Feng D, Liu J, Han Z, Zhang X (2014) Exploring permission-induced risk in android applications for malicious application detection. *IEEE Trans Inf Foren Sec* 9(11):1869–1882
43. Wang W, Xi J, Chong A, Li L (2017) Driving style classification using a semisupervised support vector machine. *IEEE Trans Hum Mach Sys* 47(5):650–660
44. Wu H, Liu B, Su W, Zhang W, Sun J (2016) Deep filter banks for Land-Use scene classification. *IEEE Geosci Remote Sens Lett* 13(12):1895–1899
45. Xu Q, Zhang C, Zhang L, Song Y (2016) The learning effect of different hidden layers stacked autoencoder. In: 2016 8th international conference on intelligent human-machine systems and cybernetics (IHMSC). IEEE, pp 148–151

Affiliations

Md Arafatur Rahman¹  · A. Taufiq Asyhari² · Ong Wei Wen¹ · Husnul Ajra³ · Yussuf Ahmed² · Farhat Anwar⁴

A. Taufiq Asyhari
taufiq-a@ieee.org

Ong Wei Wen
wei.wen96@hotmail.com

Husnul Ajra
husnul5606ice@gmail.com

Yussuf Ahmed
yussuf.ahmed@bcs.org

Farhat Anwar
farhat@iium.edu.my

¹ Faculty of Computing, University Malaysia Pahang, Pahang, Malaysia

² School of Computing and Digital Technology, Birmingham City University, Birmingham, UK

³ Department of Computer Science and Engineering, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj, Bangladesh

⁴ Faculty Engineering, International Islamic University Malaysia, Selangor, Malaysia