CCF Transactions on Networking (2020) 3:261–271
https://doi.org/10.1007/s42045-020-00040-z

REGULAR PAPER

An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks

Majd Latah[1] · Levent Toker[2]

Received: 28 July 2019 / Accepted: 4 September 2020 / Published online: 20 October 2020

Abstract

Software-defined networking (SDN) is a novel networking paradigm that provides enhanced programming abilities, which can be used to solve traditional security challenges on the basis of more efficient approaches. The most important element in the SDN paradigm is the controller, which manages the flows of each correspondence forwarding element (switch or router). Flow statistics provided by the controller are considered to be useful information that can be used to develop a network-based intrusion detection system. Therefore, in this paper, we propose a 5-level hybrid classification system based on flow statistics in order to attain an improvement in the overall accuracy of the system. For the first level, we employ the k-nearest neighbor approach (kNN); for the second level, we use the extreme learning machine (ELM); and for the remaining levels, we utilize the hierarchical extreme learning machine (HELM) approach. In comparison with conventional supervised machine learning algorithms and other state-of-the-art methodologies based on the NSL-KDD benchmark dataset, the experimental study showed that our system achieves a good accuracy (84.29%), with an ability to detect new attacks that reaches 77.18%. Therefore, our approach presents an efficient approach for intrusion detection in SDNs.

Keywords Extreme learning machine (ELM) · Hierarchical extreme learning machine (h-ELM) · Intrusion detection systems (IDS) · K-nearest neighbor (kNN) · Software-defined networking (SDN)

1 Introduction

Software-defined networking (SDN) is the fruit of earlier proposals that mainly concerned both programmable networks and the separation of control-data planes (Jarraya et al., 2014). In the SDN paradigm, the controller, which represents a logically centralized controlling point, successfully manages and gathers flow-based statistics via southbound interface such as OpenFlow protocol (McKeown et al. 2008), which is maintained by open networking foundation (ONF), a non-profit industry consortium that offers support and ensures various improvements in the SDN field (McKeown et al. 2008).

In this paper, we aim to design an efficient intrusion detection system (IDS) in the SDN paradigm. Our goal is to correctly classify well-known attacks alongside being able to detect new attacks on the basis of flow statistics provided by the controller. Flow-based intrusion detection approaches (Kim et al. 2004; Lakhina et al. 2005; Brauckhoff et al. 2007) depend only on the inspection of the packet header; therefore, they are considered to be computationally efficient in comparison with packet-based systems that require the analysis of the packet's payload (Umer et al. 2017).

In addition, packet-based systems cannot be used when network traffic is encrypted (Koch 2011). Conversely, flow-based approaches cannot be utilized when the attack is embedded into the packet's payload (Sperotto and Pras 2011). Therefore, packet and flow-based approaches can be employed together in order to achieve the highest level of protection (Sperotto et al. 2010; Golling et al. 2014). Moreover, multi-layer approaches (Xiang et al. 2004; Al-Nashif et al., 2008; Abuadlla et al. 2014; Hussain et al. 2016; Amoli and Hämäläinen 2013; Aziz et al. 2013; Cordella and Sansone 2007; Gogoi et al. 2011, 2013; Borah and Bhattacharyya 2008; Jin et al. 2006; Reddy et al. 2006; Lee et al. 2008;

✉ Majd Latah
majd.latah@ozu.edu.tr

Levent Toker
levent.toker@ege.edu.tr

[1] Department of Computer Science, Ozyegin University, 34794 Istanbul, Turkey

[2] Department of Computer Engineering, Ege University, 35100 Bornova, Turkey

Rajeswari and Kannan, 2008; Araki et al. 2014) serve as a potential solution to enhance the overall accuracy of intrusion detection systems.

In this work, we utilize a multi-layer approach based on the K-nearest neighbor (kNN), the extreme learning machine (ELM), and the hierarchical extreme learning machine (H-ELM) classification methods. Our contribution is fourfold and can be summarized in the following manner:

- Designing a multi-level hybrid approach that permits intrusion detection only on the basis of 6-flow features that can be easily obtained by a typical SDN controller.
- Employing machine learning-based classification algorithms that reduce the time required for the testing stage, such as ELM and H-ELM.
- Testing the system through the use of a standard dataset (NSL-KDD) that includes a set of new attacks that did not appear in the training dataset, in order to validate the effectiveness of our proposed system.
- Accuracy improvements, from 75.75 to 84.29%, in comparison with state-of-the-art and well-known supervised machine learning approaches that employ the same flow-based features and dataset; the results also show an ability to detect new attacks reaching to 77.18%.

The rest of this work is organized as follows. Related work of multi-level IDS is presented in Sect. 2. The backgrounds of kNN, ELM and H-ELM are introduced in Sect. 3. The proposed 5-level hybrid classification system is explained in detail in Sect. 4. The dataset used in training and testing stages is reviewed in Sect. 5. The experimental results are discussed in Sect. 6. Evaluation metrics are introduced in Sect. 7. Finally, the paper is concluded in Sect. 8.

2 Related work

In Xing et al. (2004), intrusions were classified through the use of a 3-level classification model based on C4.5 algorithm. The first stage classifies the test data into their corresponding attack group (DOS, PROBE, Others or Normal). The second stage classifies “Others” into U2R and R2L groups. The third stage categorizes the attacks into one specific attack type (e.g. back, land, etc.). The model achieved an 88.3% accuracy level with considerably high false alarm rate that reached 11.158%. Detection rate for known attacks reached 84.312%. However, it reached 42.002% for unknown attacks, which is considered to be the main limitation for this model.

MLIDS (Al-Nashif et al. 2008), on the other hand, achieved a high detection rate (99.986%) and minimized the false alarm rate to almost zero through the analysis of network traffic on the basis of three levels of granularities (rulebased flow analysis, protocol behavior analysis, and payload behavior analysis) and the utilization of an efficient fusion decision algorithm on the basis of least squares technique for each connection key and time-window. Since it includes a sliding time-window and payload analysis, it is likely to introduce a burden on the network, and therefore, it is used for the purpose of analysis.

Abuadlla et al. (2014) proposed a two-stage intrusion detection and classification method by taking advantage of two separate neural networks for each task. The first stage detects traffic anomalies whereas the next stage classifies the attacks as they occur. The experimental studies were conducted on the NSL-KDD intrusion detection dataset. In the first stage, the detection rate reached 94.2% with a false alarm of 3.4%. The second stage displayed a significantly high detection rate (99.42%) with a false alarm of 0.32%. The detection rate for known-attacks reached 99.97%, whereas it reached 78% for unknown attacks. It is worth noting that Multi-layer perceptron (MLP) with Levenberg–Marquardt takes less training time in comparison with resilient backpropagation alongside having low memory consumption in comparison with Radial Basis Function. Despite mainly using flow-based features that can easily be collected from traditional network routers through the utilization of standard protocols (NetFlow, Jflow, IPFIX) (Abuadlla et al. 2014), this approach uses in its first stage (SYN—SYN/ACK) features, where it should inspect the header of each packet in order to extract related TCP flags, and therefore, it represents an expensive feature collection that, in turn, could be an additional burden on the network.

Hussain et al. (2016) proposed a two-stage intrusion detection system with the employment of the support vector machine (SVM) as an anomaly at the first stage and the artificial neural network (ANN) as a misuse detector at the next stage. Experimental studies were conducted on the NSLKDD intrusion detection dataset. The first stage classifies the network traffic into normal and attack groups. The next stage classifies the attack traffic into four attack groups. In the first stage, the detection rate reached 99.97% with a false alarm of 0.19%. The second stage showed that the detection rate had reached 99.9% with a lower false alarm of 0.1%. The experimental results in this study reveal that misuse detection techniques are useful for the detection of known attacks with a low false positive rate.

In another study, Amoli and Hämäläinen (2013) proposed a real-time multi-stage intrusion detection system in order to enhance the detection rate of unknown attacks. The first stage detects potential attacks by monitoring a set of traffic features that include the size and number of bytes, packets, and network flows. The second stage, alternatively, finds the similarities with previous communications established by the intruders. If the number of any of the previously mentioned features crosses the threshold, a specific time slot will be flagged as anomaly. In order to increase the detection rate, the system monitors the rate of time difference between each packet (TDP) and network flows (TDF). Sub-space clustering is used to unsupervisedly cluster the previously flagged traffic, which allows the differentiation of normal traffic through the detection of the outliers with lower processing time and complexity in comparison with multidimensional clustering algorithms (Amoli and Hämäläinen 2013). After obtaining several two-dimensional clusters, DBSCAN (Ester et al. 1996) is used to create the proper clusters and to identify the outliers. The second stage is designed to discover potential similarities between intruders, which would be able to find the Bot-Master in case of Botnet attacks. To this end, past hour records of related network traffic should be stored. It is obvious that the system includes the use of multiple time windows and therefore, could negatively affect the performance of the controller if it is considered to be used in the SDN paradigm.

Aziz et al. (2013) proposed a three-stage hybrid intelligent approach. The first stage employs the principal component analysis (PCA) method for feature selection, selecting 22 out of 41 features from the NSL-KDD benchmark dataset. The second stage applies genetic algorithm in order to generate anomaly detectors that are used to identify normal and anomalous behaviors. The last layer utilizes different classifiers (Naive bayes, MLP neural network, and decision trees) in order to increase the detection accuracy. The experimental studies showed that naive bayes classifier has better accuracy in detecting U2R and R2L attacks. Conversely, the J48 decision tree classifier achieves a good accuracy (82%) in detecting DoS attacks and 65.4% in detecting probe attacks.

Cordella and Sansone (2007) proposed a serial multi-stage intrusion detection system on the basis of the learning vector quantization (LVQ) classifiers, where the authors use the reliability concept (Cordella et al., 1998) in order to assist the decision making at each stage. In case of low reliability, the system will not raise an alert which, in turn, will aid the reduction of the false alarm rate. A connection will be declared as normal traffic only if all stages do not detect any attack. The first stage distinguishes between normal traffic and the attacks belong to DoS or Probe classes. The second stage discriminates between normal traffic and R2L attacks. And finally, the last stage classifies the network traffic into normal and U2R attack classes. The overall error and missed detection reached 0.68% and 0.67% respectively.

Gogoi et al. (2013) proposed a three-level intrusion detection system that utilized a combination of supervised, unsupervised, and outlier-based techniques for the enhancement of detection accuracy. The first level classifies test data into three classes: DoS, Probe, and Rest (unclassified), on the basis of an improved version of the CatSub algorithm (Borah and Bhattacharyya 2008). The next level categorizes the Rest into Normal and Rest classes depending upon the k-point algorithm (Gogoi et al. 2011). The last level splits the Rest into U2R and R2L classes. The data of the U2R class were extracted from the Rest class through the employment of an outlier-based classifier (Jin et al. 2006). The remaining records in the Rest class were classified as R2L. The efficiency of this model was demonstrated by experimental studies conducted on four datasets, namely TUIDS (packet and flow level), DDoS, KDD Cup 99, and NSL-KDD. It is worth observing that the system was able to achieve high accuracy reaching 99.3%, 99.2%, 98.901%, 97.568%, and 98.394% on the TUIDS-packet, TUIDS-flow, DDoS, KDD Cup 99, and NSL-KDD datasets respectively. Alternatively, false positive rate on the previously mentioned datasets reached 0.0108%, 1.36%, 0.3505%, 9.933%, and 1.585% respectively.

Reddy et al. (2006) proposed a two-stage system on the basis of a combination of rule-based classifier and K-means clustering. Experimental results based on the KDD Cup 99 dataset displayed a good accuracy reaching 83.96%. Lee et al. (2008) proposed a multi-stage agent-based intrusion detection system that utilizes the Hidden Markov Model (HMM). Based on experimental studies conducted on the DARPA 2000 intrusion detection dataset, detection rates were higher than 90% and duration times were below 30 ms. Rajeswari and Kannan (2008) proposed a multi-stage system on the basis of the enhanced C4.5 algorithm. The detection rate based on the KDD Cup 99 dataset reached 62.33%, with a 9.1% false-alarm rate.

Araki et al. (2014) employed a multi-stage one-class support vector machine (OC-SVM) system to detect sophisticated unknown attacks, wherein the system makes use of three sets of traffic, two sets retrieved from a dataset and one extracted from the real network. At the first stage, the OC-SVM learns from older archival data and later analyzes a newer dataset and one from the real network. At the second stage, the OC-SVM learns the outliers from the newer set and utilizes it in order to analyze the outliers of the real network. The detection rate and the false positive rate reached 80.00% and 20.94% respectively.

Casas et al. (2011) proposed a three-step clustering techinque on the basis of sub-space density clustering in order to detect outlying flows where they use the following features: number of source/destination IP addresses and ports, ratio of number of sources to number of destinations, packet rate, fraction of ICMP and SYN packets, and average packet size. The first step detects anomalous time slots in which the analysis be conducted. The next step identifies and ranks the outlying flows according to the degree of abnomality on the basis of a combination of sub-space clustering (Parsons et al. 2004), DBSCAN (Ester et al. 1996), and evidence accumulation clustering (Fred and Jain 2005).

Thereafter, the top-ranked flows are considered as anomalies on the basis of a thresholding apporach. The approach proposed by Casas et al. (2011) displayed a better performance in comparison with other unsupervised anomaly detection methods such as DBSCAN, k-means, and PCA. However, since it extracts some TCP flags such as SYN, it may not be appropriate for the SDN case due to the fact that dealing with flows is more faster and efficient than handling each packet separately. Previously discussed studies depend mainly on the use of the whole dataset that probably contains different features (basic features, time-related features, connection-related features, content-related features, host-related features, and login attempts-related features). Al-Yaseen et al. (2017) proposed a five-level classification model based on a combination of SVM and ELM algorithms. The experimental results conducted on the KDD Cup 99 dataset, showed that the system was able to achieve a high accuracy (95.75%) with low false alarm rate (1.87%).

Using all these features, however, may not be the best choice for designing a network-based IDS in the SDN paradigm. Recently, a flow-based deep learning approach (Tang et al. 2016) has been proposed for the purpose of intrusion detection in SDNs, where the system achieved a good accuracy reaching 75.75% only on the basis of 6-flow features. The same features also are used in our proposed system. The same authors also (Tang et al. 2018) improved the results of the previous study by applying gated recurrent unit recurrent neural network (GRU-RNN) approach. The authors were able to achieve an accuracy of 89% with the same six raw features.

Wang et al. (2018) adopted a semi-supervised approach combined with contrastive pessimistic likelihood estimation (CPLE) for real-time detection in a wireless SDN environment. The semi-supervised approach trained the model with both labeled and unlabeled data. The experimental results showed that this approach can outperform (Tang et al. 2016) with an accuracy of 77.26%, again using the same six raw features of SDN. In addition, this approach is more realistic since it utilizes the unlabeled data, which does not require any additional cost for further expert-based labelling (Latah and Toker 2018a).

On the other hand, Latah and Toker (2018b) showed that it is possible to achieve good accuracy and higher detection by excluding the content features of NSL-KDD and then applying feature selection using PCA approach. This work achieved an accuracy of 88.74% and also outperformed (Tang et al. 2016; Wang et al. 2018) in terms of accuracy as well as Tang et al. (2016, 2018) in terms of recall (96.50%) and F1-score (89.38%). In the same context, Dey et al. (2018) compared different supervised machine learning approaches along with common feature selection methods for predicting possible intrusions in SDNs. The experimental results showed that random forest approach, combined with information gain as a feature selection step, was able to attain an accuracy of 81.95%.

The usage of flow-based features is considered straightforward and very effective in the SDN architecture where the SDN controller is already able to obtain such features without any need for a preprocessing or flow aggregation step. In this study, we investigate multi-level flow-based intrusion detection for SDNs. We also compare our results with conventional supervised learning approaches and other related works using the same dataset.

3 Theoretical background

In this section, we introduce the theoretical background of the algorithms used in our proposed system. Therefore, we describe in details the following algorithms: kNN, ELM, and H-ELM.

3.1 K-nearest neighbor algorithm (kNN)

K-nearest neighbor algorithm (kNN) classifies the new instances that exist in a given dataset according to their closest training instances in the feature space (Dasarathy 1991). kNN is a straightforward algorithm and displays a good robustness against noisy training data or a large dataset (Bhattacharya et al. 2012). It is worth mentioning that kNN and its variants (Li and Guo 2007; Kuang and Zulkernine 2008; Holmes and Adams 2002; Popescu and Keller 2016; Sperotto et al. 2010) are used widely in malware detection (Alazab et al. 2011; Santos et al. 2009), intrusion detection (Li et al. 2014; Damopoulos et al. 2012; Liao and Vemuri a, 2002b), and spam detection (Firte et al. 2010). Typically, the algorithm finds the k closest instances based on a calculation of the distance between the new instances and all training instances. For instance, let X and Y be two feature vectors with n dimension. The Euclidian distance between these two feature vectors is defined below

$$d_{(X,Y)} = \sum_{i=0}^{n} \left(X_i - Y_i\right)^2 \tag{1}$$

The new instance is classified based on the majority vote of its nearest instances. Therefore, it will be assigned to the class whose labels are the most frequent. Other distance metrics such as Manhattan, Mahalanobis, Chebyshev, Minkowski, and Hamming can be utilized as well. The performance of kNN is affected by the distance metric used by the algorithm alongside the choice of the optimal value of the parameter k (Parvin et al. 2010). A small k increases the impact of individual cases. A large k, however, increases the robustness of the noise provided in the dataset (Anbeek et al. 2004). The value of k is commonly determined through the employment of cross-validation or adaptive approaches (Wettschereck and Dietterich 1994). In this study, we use the standard version of this algorithm by using the Euclidian distance as the distance metric and applying cross-validation in order to determine the optimal value of parameter k. kNN algorithm is used in the first layer of our proposed model in order model for the detection of DoS attacks.

3.2 Extreme learning machine (ELM)

The extreme learning machine (ELM) (Huang et al. 2004) represents a new learning paradigm built on basis of the concept of single-hidden layer-feedforward neural network. ELM requires short training and testing time in comparison with traditional feed-forward networks owing to the fact that the parameters of its hidden layer are randomly chosen, thereby eliminating the need for the training stage (Al-Yaseen et al. 2017). ELM achieved high accuracy for multi-level IDS as compared with SVM (Cheng et al. 2012). Kernel based ELM outperformed neural networks approaches for detecting Probe attacks in a multi-level IDS model (Singh et al., 2015). For a given data set $Z\{(x_1,t_1),(x_2,t_2),\dots,(x_i,t_i) : i=1,\dots,N\}$ where $x_i=[x_{i1},x_{i2},\dots,x_{in}]^T\in R^n$ and $t_i=[t_{i1},t_{i2},\dots,t_{im}]^T\in R^m$ which is the label of instances, a single-hidden layer feed-forward neural network (SLFN) with n inputs, m outputs, k hidden neurons and activation function $g(x)$ can be written as

$$\sum_{i=1}^{k}\beta_i g\left(w_i^T x_j+b_i\right)=t,\qquad j=1,\dots,N \tag{2}$$

where $W_i=[w_{i1},w_{i2},\dots,w_{in}]^T$ is the weight vector between the i_{th} hidden neuron and the input neurons, $\beta_i=[\beta_{i1},\beta_{i2},\dots,\beta_{in}]^T$ is the weight vector between the i_{th} hidden neuron and the outputs, and b_i is the threshold of the i_{th} hidden neuron. The above equation can be written as

$$\boldsymbol{H}\beta=T \tag{3}$$

where

$$\boldsymbol{H}=\begin{bmatrix} g\left(w_1^T x_1+b_1\right) & \cdots & g\left(w_k^T x_1+b_k\right)\\ \vdots & \ddots & \vdots\\ g\left(w_1^T x_N+b_1\right) & \cdots & g\left(w_k^T x_N+b_k\right)\end{bmatrix}$$

$$\beta=\begin{bmatrix}\beta_1^T\\ \vdots\\ \beta_k^T\end{bmatrix} \text{ and } T=\begin{bmatrix}t_1^T\\ \vdots\\ t_k^T\end{bmatrix}$$

The output weights of ELM are obtained by determining a solution that achieves the least squares error of the linear system. The unique smallest norm least-squares solution can be obtained by

$$\hat{\beta}=\boldsymbol{H}^{\dagger}\ T \tag{4}$$

where $H^{\dagger}$ represents the Moore–Penrose generalized inverse of matrix H. The Moore–Penrose generalized inverse can be calculated by utilizing different approaches such as the orthogonal projection method, iterative method, and singular value decomposition (Banerjee 1973). On the basis of ridge regression theory and the fact that the number of training samples is far bigger than the dimensionality of the feature space, it is better to add a positive value to the diagonal of $H^T T$, which provides a better generalization performance as shown numerically in Huang et al. (2011).

$$\hat{\beta}=\left(\frac{\boldsymbol{I}}{\boldsymbol{C}}+\boldsymbol{H}^T\mathbf{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{T} \tag{5}$$

where C is a positive constant. For binary classification, the decision function can be written in the manner shown below

$$\mathrm{f(x)}=sign\left(h(x)\left(\frac{\boldsymbol{I}}{\boldsymbol{C}}+\boldsymbol{H}^T\mathbf{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{T}\right) \tag{6}$$

ELM is used in the second layer of our multi-level system to efficiently detect the Probe attacks.

3.3 Hierarchical Extreme Learning Machine (H-ELM)

The Hierarchical Extreme Learning Machine (H-ELM) (Tang et al. 2015) was proposed in order to achieve a better generalization with faster convergence in comparison with the fundamental ELM approach. H-ELM training is structurally divided into two stages: (1) unsupervised hierarchical feature representation and (2) supervised feature classification. First, the input instances should be converted into an ELM feature space, which may help in the extraction of hidden information from training instances. Then, high-level sparse features are obtained by applying a M-layer unsupervised learning stage. The output of each hidden layer is defined as

$$\boldsymbol{H_i}=g(\boldsymbol{H}_{i-1}\cdot\beta) \tag{7}$$

where H_i is the output of the hidden layer i_{th}, i∈[1, k], H_{i-1} is the output of the $(i-1)_{th}$ hidden layer, g(·) represents the activation function of the hidden layers, and β denotes the output weights. After extracting the features of the previous layer, the parameters of the current hidden layer will remain fixed (Tang et al. 2015). H-ELM employs random projections of extracted features as the inputs of the feature classification stage. ℓ_1 penalty is applied to produce more sparse and significant information. The input weights of the ELM sparse autoencoder are obtained by searching the path from a random mapped feature space (Tang et al. 2015). The autoencoder enjoys universal approximation capability and sparse constraint is added to the optimization model, which is written as

$$O_{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \|\boldsymbol{H}\beta - X\|^2 + \|\beta\|_{\ell_1} \right\} \tag{8}$$

where X denotes the input data, H represents the random mapping output, and β is the hidden layer weight. A fast iterative shrinkage-thresholding algorithm (FISTA) (Beck and Teboulle 2009) is used in order to solve Eq. (6). As mentioned before, the supervised training stage is implemented by the original ELM, which is already explained in Sect. 3.2. H-ELM is used in the third, fourth, and fifth layers of our proposed system in order to detect U2R, R2L, and unknown attacks respectively.

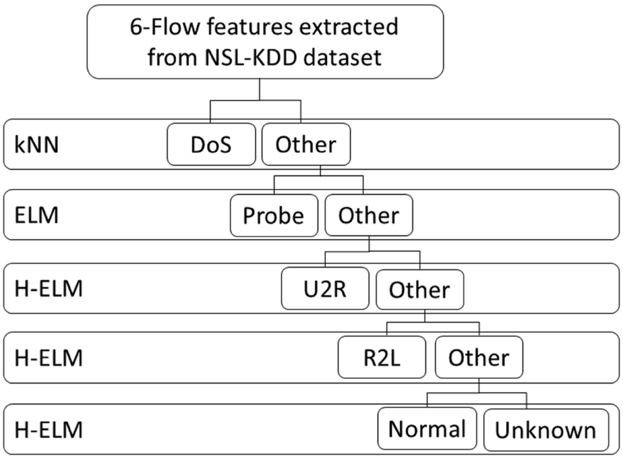

Fig. 2 The architecture of our proposed IDS

4 Proposed multi-level hybrid IDS

In this section, we present our proposed system. The system is inspired by the study presented in Al-Yassen et al. (2017), which uses a five-level classification model based on a combination of SVM and ELM algorithms. As shown in Fig. 1, DoS and Probe attacks are detected prior to other categories. This is due to the fact that both DoS and Probe attacks have lower similarity in comparison with other attack types (Gogoi et al. 2013). Conversely, both U2R and R2L attacks are relatively similar to the normal traffic patterns (Gogoi et al. 2013) and therefore, both of them present a potential threat (Sharma and Mukherjee 2012).

Similar to the model presented in Al-Yassen et al. (2017), our model uses one classifier per layer. As shown in Fig. 2, the second layer remains the same where ELM achieves better performance as compared to other approaches such as SVM and neural networks (Cheng et al. 2012; Singh et al. 2015; Banerjee 1973). In the next layers, we replace the SVM with H-ELM, which is faster than SVM and performs a better generalization in comparison with SVM and other approaches (Al-Yaseen et al. 2017; Cheng et al. 2012; Singh et al. 2015). It is worth noting that in our model we use 6-flow features instead of the 41 traffic-features used in Cheng et al. (2012), not to mention that our experimental study is conducted on the basis of the NSL-KDD, which is an enhanced version of the KDD Cup 99. At each layer, the training dataset is spilt into two categories: the first one represents a specific attack type and “Other” category, which represents a normal traffic or an attack that has to be detected by the next layer. Therefore, the model uses five datasets and utilizes the one-versus-all training approach. In the first layer, we use the kNN approach instead of the SVM due to its robustness against noisy training data (Bhattacharya et al. 2012).

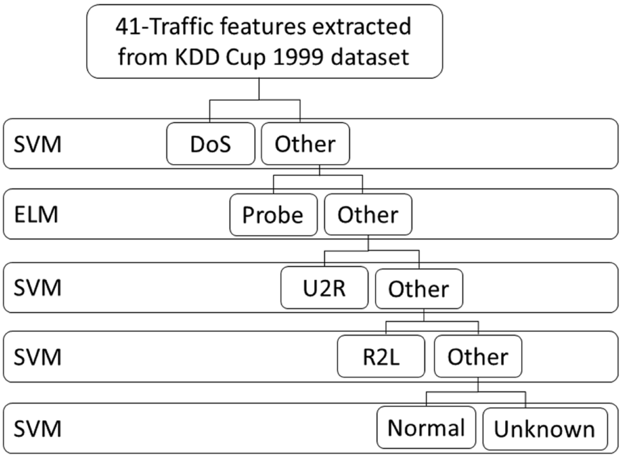

Fig. 1 Multi-level hybrid IDS proposed in Al-Yassen et al. (2017)

5 Dataset and selected features

As mentioned in the previous sections, in this study, we use the NSL-KDD dataset. The NSL-KDD is an enhanced version of the KDD Cup 99 dataset that suffers from a huge number of redundant records (Tavallaee et al. 2009). The NSL-KDD dataset includes the features shown in Table 1. We use features number F1, F2, F5, F6, F23, and F24, which can be easily obtained from the SDN controller (Tang et al. 2016).

The NSL-KDD dataset contains a total of 39 attacks wherein each attack is classified into one of the following four categories. In addition, a set of these attacks is added to the testing set. Table 2 presents the distribution of the known and new attack records in the NSL-KDD testing set.

Table 1 List of features of NSL-KDD dataset

F. #	Feature name	F. #	Feature name	F. #	Feature name
F1	Duration	F15	Su attempted	F29	Same srv rate
F2	Protocol type	F16	Num root	F30	Diff srv rate
F3	Service	F17	Num file creations	F31	Srv diff host rate
F4	Flag	F18	Num shells	F32	Dst host count
F5	Source bytes	F19	Num access files	F33	Dst host srv count
F6	Destination bytes	F20	Num outbound cmds	F34	Dst host same srv rate
F7	Land	F21	Is host login	F35	Dst host diff srv rate
F8	Wrong fragment	F22	Is guest login	F36	Dst host same src port rate
F9	Urgent	F23	Count	F37	Dst host srv diff host rate
F10	Hot	F24	Srv count	F38	Dst host serror rate
F11	Number failed logins	F25	Serror rate	F39	Dst host srv serror rate
F12	Logged in	F26	Srv serror rate	F40	Dst host rerror rate
F13	Num compromised	F27	Rerror rate	F41	Dst host srv rerror rate
F14	Root shell	F28	Srv rerror rate	F42	Class label

Table 2 Distributions of known and new attacks in KDD-Test set

	DoS	R2L	U2R	Probe
Known attacks	5741	2199	37	1106
	76.98%	79.85%	18.50%	45.68%
New attacks	1717	555	163	1315
	23.02%	20.15%	81.50%	54.32%

Table 3 Parameters used for each classifier at each layer

Layer	Classifier	Parameter		
1	kNN	K = 65		
2	ELM	N = 400		
3	H-ELM	N1	N2	N3
		40	40	200
4	H-ELM	N1	N2	N3
		10	10	300
5	H-ELM	N1	N2	N3
		10	10	200

6 Evaluation metrics

The performance of our multi-level flow-based system is evaluated in terms of accuracy and False Alarm Rate (FAR) where the accuracy is calculated by

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

True positive (TP) is the number of attack instances correctly classified; true negative (TN) is the number of normal traffic instances correctly classified; false positive (FP) is the number of normal traffic instances falsely classified; and false negative (FN) is number of attack instances falsely classified. The false alarm rate is calculated by

$$False\,Alarm\,Rate = \frac{FP}{TN + FP} \tag{10}$$

In addition, we calculate Precision, Recall and F-measure which are obtained by

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall\ (Detection\,Rate) = \frac{TP}{TP + FN} \tag{12}$$

$$F - measure = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{13}$$

Precision reveals the percentage of attacks detected by an IDS that are actual attacks. Recall indicates the percentage of detected attacks versus all attacks presented in the NSL-KDD dataset. F-measure represents a more balanced measure by considering both precision and recall (Tang et al. 2016).

7 Experimental results

The experiment study was conducted on an Intel i5 machine with 12 GB of RAM. Table 3 shows the parameters used for each classifier employed at each layer.

We divided our experimental study into two experiments: (1) evaluation of the performance at each layer and (2) comparison with other supervised machine learning approaches. The first experiment basically evaluated the performance of the model. Table 4 shows the accuracy and the false alarm rate achieved at each layer.

Table 4 Results for training and testing stages achieved at each layer of our proposed system

Layer #	Classifier	Detected attack	Training stage		Testing stage	
			Accuracy (%)	False alarm rate (%)	Accuracy (%)	False alarm rate (%)
1	kNN	DoS	97.34	1.38	91.23	3.39
2	ELM	Probe	97.12	0.12	92.61	1.45
3	H-ELM	U2R	99.96	0.0024	99	1.1
4	H-ELM	R2L	99.19	0.0240	86.97	0.94
5	H-ELM	Unknown attacks	90.76	7.28	80.39	5.61
All layers	kNN + ELM + H-ELM	All types	94.11	7.89	84.29	6.3

In the second experiment, we compared our proposed system with Tang et al. (2016, 2018); Latah and Toker (2018b); Dey et al. (2018); Wang et al. (2018) and other conventional supervised learning approaches. As displayed in Table 5, compared with conventional supervised machine learning approaches our system achieves the highest values of accuracy, recall, and F1-score. In terms of false positive rate, it is worth mentioning that H-ELM approach achieves the highest precision with the lowest false alarm rate. On the other hand, the detection rate for new attacks in our proposed system reached 77.18%. Our approach also outperformed the semi-supervised learning approach proposed by Wang et al. (2018) and the supervised learning approach introduced by Dey et al. (2018) which combines conventional supervised machine learning approaches along with commonly used feature selection methods.

Compared with (Tang et al. 2016), in which the authors used a simple deep neural network with an input layer, three hidden layers and an output layer, our system was able to achieve a higher accuracy, precision, recall, and F1-score. However, in terms of false alarm rate, Tang et al. (2016) showed better results. Tang et al. (2018) were able to improve the accuracy, recall and F1-score compared with their previous study (Tang et al. 2016), Dey et al. (2018) and our proposed system. This is due to the fact that GRU-RNN can represent the relationship between current and previous events and consequently will enhance the detection rate. Latah and Toker (2018b) achieved the best recall and F1-score; however, they applied PCA as a feature selection method to improve the overall performance of the system. This study also shows that PCA can achieve better results compared with commonly used feature selection approaches such as information gain used in Dey et al. (2018).

Table 5 Comparing our proposed approach with other approaches

Method	Feature selection	Accuracy (%)	False alarm rate (%)	Precision (%)	Recall (%)	F1-score (%)
Naive Bayes	–	49.88	5.14	80.28	15.83	26.45
Neural Network	–	63.70	7.66	87.88	42.02	56.86
SVM	–	71.40	10.63	87.79	57.80	69.71
Decision Tree	–	74.43	6.43	92.50	59.95	72.75
ELM (N = 1500)	–	74.80	10.17	89.18	63.43	74.13
kNN	–	77.09	4.07	95.33	62.84	75.75
H-ELM (N1 = 30,N2 = 30,N3 = 300)	–	77.59	2.57	96.98	62.59	76.08
H-ELM (N1 = 10,N2 = 10,N3 = 200)	–	80.39	5.61	94.26	69.80	80.21
Simple Deep Neural Network (Tang et al. 2016)	–	75.75	3.21	92.50	59.95	74.13
Semi-supervised Approach (Wang et al. 2018)	–	77.26	N.A	N.A	N.A	N.A
Random Forest (Dey et al. 2018)	Information gain	81.95	N.A	N.A	N.A	N.A
Our approach	–	84.29	6.3	94.18	77.18	84.83
Decision Tree (Latah and Toker 2018b)	PCA	88.74	3.99	83.24	96.5	89.38
GRU-RNN (Tang et al. 2018)	–	89	N.A	89	89.5	89.2

Approaches with (–) sign use the same six raw features (i.e., F1, F2, F5, F6, F23, and F24).

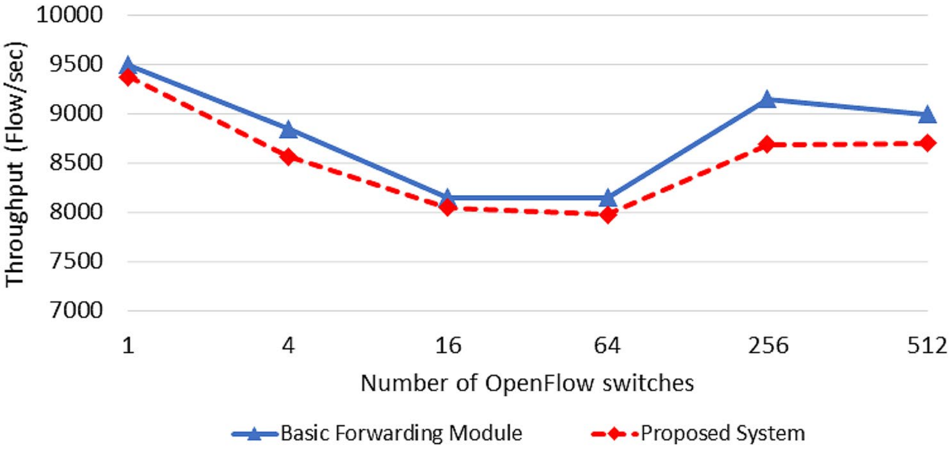

Fig. 3 Comparing between the throughput of our proposed IDS and basic forwarding module

We also used the Cbench tool in order to evaluate the throughput of our system. Therefore, we implemented the system as a module of POX controller in the SDN's control plane. This approach is considered more efficient than the implementation of the system as an application of the controller due to the fact that our 6-flow features can be easily obtained by the controller, and increasing number of flows will dramatically increase the interaction between the controller and the application. Previously mentioned features were collected periodically every 10 seconds. Each Open vSwich was connected to 1000 virtual hosts with different MAC addresses, sending 10,000 (Packet-In) messages to the POX controller. Compared with the basic forwarding module, our system achieved an acceptable performance as presented in Fig. 3. This is owing to the fact that we use only 6-flow features that can easily be obtained from the controller and employ machine learning algorithms that reduce the time required for the testing stage such as ELM and H-ELM in the layers from 2 to 5.

In brief, our experimental study showed that multi-level IDS can outperform common supervised machine learning approaches as well as other approaches (Tang et al. 2016; Wang et al. 2018; Dey et al. 2018). However, more advanced techniques such as GRU-RNN (Tang et al. 2018) were able to achieve more promising results. Second, multi-level intrusion detection systems need to be improved in order to decrease the number of false alarms, which is observed in our proposed system.

8 Conclusion

In this paper, we proposed an efficient multi-level hybrid intrusion detection method for SDNs. The system was designed on the basis of a combination of kNN, ELM, and H-ELM approaches. The experimental study conducted on NSL-KDD dataset showed that our approach significantly enhanced the overall accuracy when compared with conventional supervised learning approaches. In addition, the system was able to detect the new attacks included in the testing set with a detection rate that reached to 77.18%. Our future work will be focused on improving the system in order to achieve a lower false alarm rate.

Acknowledgements We would like thank the anonymous reviewers for their insightful comments and constructive suggestions, which helped us improve the quality of this work.

References

Abuadlla, Y., Kvascev, G., Gajin, S., Jovanovic, Z.: Flow-based anomaly intrusion detection system using two neural network stages. Comput. Sci. Inf. Syst. **11**(2), 601–622 (2014)

Alazab, M., Venkatraman, S., Watters, P., Alazab, M.: Zero-day malware detection based on supervised learning algorithms of api call signatures. In: Proceedings of the Ninth Australasian Data Mining Conference-Volume 121, pp. 171–182. Australian Computer Society, Inc. (2011)

Al-Nashif, Y., Kumar, A.A., Hariri, S., Luo, Y., Szidarovsky, F., Qu, G.: Multilevel intrusion detection system (ml-ids). In: 2008 International Conference on Autonomic Computing, pp. 131–140. IEEE (2008)

Al-Yaseen, W.L., Othman, Z.A., Nazri, M.Z.A.: Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system. Expert Syst. Appl. **67**, 296–303 (2017)

Amoli, P.V., Hämäläinen, T.: Real time multi stage unsupervised intelligent engine for nids to enhance detection rate of unknown attacks. In: 2013 IEEE Third International Conference on Information Science and Technology (ICIST), pp. 702–706. IEEE (2013)

Anbeek, P., Vincken, K.L., Van Osch, M.J., Bisschops, R.H., Van Der Grond, J.: Probabilistic segmentation of white matter lesions in mr imaging. NeuroImage **21**(3), 1037–1044 (2004)

Araki, S., Yamaguchi, Y., Shimada, H., Takakura, H.: Unknown attack detection by multistage one-class svm focusing on communication interval. In: International Conference on Neural Information Processing, pp. 325–332. Springer (2014)

Aziz, A.S.A., Hassanien, A.E., Hanaf, S.E.O., Tolba, M.F.: Multi-layer hybrid machine learning techniques for anomalies detection and classification approach. In: 13th International Conference on Hybrid Intelligent Systems (HIS 2013), pp. 215–220. IEEE (2013)

Banerjee, K.: Generalized inverse of matrices and its applications. (1973)

Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)

Bhattacharya, G., Ghosh, K., Chowdhury, A.S.: An affinity-based new local distance function and similarity measure for knn algorithm. Pattern Recognit. Lett. **33**(3), 356–363 (2012)

Borah, B., Bhattacharyya, D.: Catsub: a technique for clustering categorical data based on subspace. ICFAI J. Comput. Sci. 7–20 (2008)

Brauckhoff, D., May, M., Plattner, B.: Flow-level anomaly detection - blessing or curse? In: IEEE INFOCOM 2007, Student Workshop, Anchorage, Alaska, USA (May 2007)

Casas, P., Mazel, J., Owezarski, P.: Unada: unsupervised network anomaly detection using sub-space outliers ranking. In: International Conference on Research in Networking, pp. 40–51. Springer (2011)

Cheng, C., Tay, W.P., Huang, G.: Extreme learning machines for intrusion detection. In: The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2012)

Cordella, L., Sansone, C., Tortorella, F., Vento, M., De Stefano, C.: Neural network classification reliability: problems and applications. Image Process. Pattern Recognit. **5**, 161–200 (1998)

Cordella, L.P., Sansone, C.: A multi-stage classification system for detecting intrusions in computer networks. Pattern Anal. Appl. **10**(2), 83–100 (2007)

Damopoulos, D., Menesidou, S.A., Kambourakis, G., Papadaki, M., Clarke, N., Gritzalis, S.: Evaluation of anomaly-based ids for mobile devices using machine learning classifiers. Secur. Commun. Netw. **5**(1), 3–14 (2012)

Dasarathy, B.V.: Nearest neighbor (nn) norms: Nn pattern classification techniques. IEEE Comput. Soc. Tutor. (1991)

Dey, S.K., Uddin, M.R., Rahman, M.M.: Detection of flow based anomaly in OpenFlow controller: machine learning approach in software defined networking. In: 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), pp. 416–421 (2018).

Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd **96**, 226–231 (1996)

Firte, L., Lemnaru, C., Potolea, R.: Spam detection filter using knn algorithm and resampling. In: Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing, pp. 27–33. IEEE (2010)

Fred, A.L., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 835–850 (2005)

Gogoi, P., Borah, B., Bhattacharyya, D.K.: Network anomaly detection using unsupervised model. Int. J. Comput. Appl. (Special Issue on Network Security and Cryptography) NSC, 19–30 (2011)

Gogoi, P., Bhattacharyya, D., Borah, B., Kalita, J.K.: Mlh-ids: a multi-level hybrid intrusion detection method. Comput. J. **57**(4), 602–623 (2013)

Golling, M., Hofstede, R., Koch, R.: Towards multi-layered intrusion detection in high-speed networks. In: 2014 6th International Conference on Cyber Conflict (CyCon 2014), pp. 191–206. IEEE (2014)

Holmes, C., Adams, N.: A probabilistic nearest neighbour method for statistical pattern recognition. J. R. Stat. Soc. Ser. B (Statistical Methodology) **64**(2), 295–306 (2002)

Huang, G.B., Zhu, Q.Y., Siew, C.K., et al.: Extreme learning machine: a new learning scheme of feedforward neural networks. Neural Netw. **2**, 985–990 (2004)

Huang, G.B., Wang, D.H., Lan, Y.: Extreme learning machines: a survey. Int. J. Mach. Learn. Cybern. **2**(2), 107–122 (2011)

Hussain, J., Lalmuanawma, S., Chhakchhuak, L.: A two-stage hybrid classification technique for network intrusion detection system. Int. J.Comput. Intell. Syst. **9**(5), 863–875 (2016)

Jarraya, Y., Madi, T., Debbabi, M.: A survey and a layered taxonomy of software-defined networking. IEEE Commun. Surveys Tutor. **16**(4), 1955–1980 (2014)

Jin, W., Tung, A.K., Han, J., Wang, W.: Ranking outliers using symmetric neighborhood relationship. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 577–593. Springer (2006)

Kim, M.S., Kong, H.J., Hong, S.C., Chung, S.H., Hong, J.W.: A flow-based method for abnormal network traffic detection. In: 2004 IEEE/IFIP network operations and management symposium (IEEE Cat. No. 04CH37507), vol. 1, pp. 599–612. IEEE (2004)

Koch, R.: Towards next-generation intrusion detection. In: 2011 3rd International Conference on Cyber Conflict, pp. 1–18. IEEE (2011)

Kuang, L., Zulkernine, M.: An anomaly intrusion detection method using the csi-knn algorithm. In: Proceedings of the 2008 ACM symposium on Applied computing, pp. 921–926. ACM (2008)

Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: ACM SIGCOMM Computer Communication Review, vol. 35, pp. 217–228. ACM (2005)

Latah, M., Toker, L.: Artificial intelligence enabled software-defined networking: a comprehensive overview. IET Netw. **8**(2), 79–99 (2018a)

Latah, M., Toker, L.: Towards an efficient anomaly-based intrusion detection for software-defined networks. IET Netw. **7**(6), 453–459 (2018b)

Lee, D.H., Kim, D.Y., Jung, J.I.: Multi-stage intrusion detection system using hidden markov model algorithm. In: 2008 International Conference on Information Science and Security (ICISS 2008), pp. 72–77. IEEE (2008)

Li, Y., Guo, L.: An active learning based tcm-knn algorithm for supervised network intrusion detection. Comput. Secur. **26**(7–8), 459–467 (2007)

Li, W., Yi, P., Wu, Y., Pan, L., Li, J.: A new intrusion detection system based on knn classification algorithm in wireless sensor network. J. Electr. Comput. Eng. (2014)

Liao, Y., Vemuri, V.R.: Use of k-nearest neighbor classifier for intrusion detection. Comput. Secur. **21**(5), 439–448 (2002a)

Liao, Y., Vemuri, V.R.: Using text categorization techniques for intrusion detection. USENIX Secur. Symp. **12**, 51–59 (2002b)

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: Openflow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008)

Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: a review. ACM SIGKDD Explor. Newsl **6**(1), 90–105 (2004)

Parvin, H.; Alizadeh, H.; Minaes-Bidgoli, B.: MKNN: modified k-nearest neighbor. In: Proceedings of World Congress on Engineering and Computer Science (WCECS), Yantai, China, pp. 91–94 (2010)

Popescu, M., Keller, J.M.: Random projections fuzzy k-nearest neighbor (rpfknn) for big data classification. In: 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1813–1817. IEEE (2016)

Rajeswari, L.P., Kannan, A.: An intrusion detection system based on multiple level hybrid classifier using enhanced c4. 5. In: 2008 International Conference on Signal Processing, Communications and Networking, pp. 75–79. IEEE (2008)

Reddy, N.S., Acharya, U.D., et al.: A two-stage hybrid model for intrusion detection. In: 2006 International Conference on Advanced Computing and Communications, pp. 163–165. IEEE (2006)

Santos, I., Penya, Y.K., Devesa, J., Bringas, P.G.: N-grams-based file signatures for malware detection. ICEIS **2**(9), 317–320 (2009)

Sharma, N., Mukherjee, S.: A novel multi-classifier layered approach to improve minority attack detection in ids. Proc. Technol. **6**, 913–921 (2012)

Singh, R., Kumar, H., Singla, R.: An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Expert Syst. Appl. **42**(22), 8609–8624 (2015)

Sperotto, A., Pras, A.: Flow-based intrusion detection. In: 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops, pp. 958–963. IEEE (2011)

Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., Stiller, B.: An overview of ip flow-based intrusion detection. IEEE Commun. Surveys Tutor. **12**(3), 343–356 (2010)

Tang, J., Deng, C., Huang, G.B.: Extreme learning machine for multilayer perceptron. IEEE Trans. Neural Netw. Learn. Syst. **27**(4), 809–821 (2015)

Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M.: Deep learning approach for network intrusion detection in software defined networking. In: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 258–263. IEEE (2016)

Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M.: Deep recurrent neural network for intrusion detection in sdn-based networks. In 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), pp. 202–206. IEEE (2018)

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6. IEEE (2009)

Umer, M.F., Sher, M., Bi, Y.: Flow-based intrusion detection: TEchniques and challenges. Comput. Secur. **70**, 238–254 (2017)

Wang, B., Sun, Y., Yuan, C., Xu, X.: LESLA: A smart solution for SDN-enabled mMTC E-health monitoring system. In Proceedings of the 8th ACM MobiHoc 2018 Workshop on Pervasive Wireless Healthcare Workshop, pp. 1–6. IEEE (2018)

Wettschereck, D., Dietterich, T.G.: Locally adaptive nearest neighbor algorithms. In: Advances in Neural Information Processing Systems, pp. 184–191 (1994)

Xiang, C., Chong, M., Zhu, H.: Design of mnitiple-level tree classifiers for intrusion detection system. In: IEEE Conference on Cybernetics and Intelligent Systems, 2004., vol. 2, pp. 873–878. IEEE (2004)

Majd Latah received his MSc degree in Computer Engineering from Ege University, Turkey, in 2018. He is currently pursuing his PhD in Computer Science at Ozyegin University, Turkey. His research interests include network security, blockchain and software-defined networks.

Levent Toker received his MSc and PhD degrees in Computer Engineering from Ege University, Turkey, in 1988 and 1991, respectively. Since 2001, he has been a full Professor in the Department of Computer Engineering, Ege University, Turkey. His research interests include computer networks, mobile networks and network management.