GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685745

Exploiting Ensemble Learning for Edge-assisted Anomaly Detection Scheme in e-healthcare System

Wei Yao*, Kuan Zhang†, Chong Yu†, Hai Zhao*
*School of Computer Science and Engineering, Northeastern University, Shenyang, China
†Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, Omaha, USA
Email:{yaow.neu@gmail.com, kuan.zhang@unl.edu, cyu6@huskers.unl.edu, zhaoh@mail.neu.edu.cn}

***Abstract*—With the thriving of wearable devices and the widespread use of smartphones, the e-healthcare system emerges to cope with the high demand of health services. However, this integrated smart health system is vulnerable to various attacks, including intrusion attacks. Traditional detection schemes generally lack the classifier diversity to identify attacks in complex scenarios that contain a small amount of training data. Moreover, the use of cloud-based attack detection may result in higher detection latency. In this paper, we propose an Edge-assisted Anomaly Detection (EAD) scheme to detect malicious attacks. Specifically, we first identify four types of attackers according to their attacking capabilities. To distinguish attacks from normal behaviors, we then propose a wrapper feature selection method. This selection method eliminates the impact of irrelevant and redundant features so that the detection accuracy can be improved. Moreover, we investigate the diversity of classifiers and exploit ensemble learning to improve the detection rate. To reduce high detection latency in the cloud, edge nodes are used to concurrently implement the proposed lightweight scheme. We evaluate the EAD performance based on two real-world datasets, i.e., NSL-KDD and UNSW-NB15 datasets. The simulation results show that the EAD outperforms other state-of-the-art methods in terms of accuracy, detection rate, and computational complexity. The analysis of detection time validates the fast detection of the proposed EAD compared with cloud-assisted schemes.**

***Index Terms*—Anomaly Detection, e-healthcare, Feature Selection, Ensemble Learning, Edge Nodes**

I. Introduction

The e-healthcare system connects wearable sensors, smartphones, and cloud servers to support smart health services and remote personal care. It can provide users with continuous monitoring, self-management of their own health record, and emergency assistance through user's mobile devices. From recent healthcare report, the e-healthcare industry in the United States is expected to soar to over $175 billion by 2026, as more and more medical facilities adopt e-healthcare systems and smartphone-based health apps [1]. Through the health apps, smartphones can seamlessly connect with wearable devices (e.g., smart watches, bracelets, rings, and ECG monitors) to receive users' health data. Although smartphones can preprocess data, a large portion of data is transmitted to cloud servers for complicated processing. The cloud servers extract useful information from the raw data and analyze it for users' health monitoring and diagnosis. At the same time, doctors and/or their families can use smartphones to remotely access these health records and conduct assessments. Finally, the diagnostic results are sent back to the user's smartphone.

Although health applications are promising, the e-healthcare system faces various security vulnerabilities. In order to compromise e-healthcare system, attackers may exploit any vulnerabilities inside the whole system to launch attacks [2]. For example, attackers who are unauthorized may steal patients' private data from the cloud, which leads to the destruction of the medical database and ultimately damages the entire system. Malicious attackers may also fake their identities to mislead users. In such a case, doctors may face difficulties to retrieve accurate health data due to data tampering, which undermines the validity of diagnosis and causes severe consequences.

Existing detection schemes rely on statistical models [3], [4] knowledge-based rules [5], data mining and machine learning techniques [6-9]. Some related works investigated the network features and build profiles from normal patterns, while others construct classifiers for attack detection. However, building a comprehensive profile of all possible normal observations is a key challenge due to the generated heterogeneous data from the dynamics of network traffic data. Thus, existing detection schemes predominantly suffer low detection rates. In addition, it is challenging to find a single classifier to detect malicious activities effectively. This is because training an individual classifier on different network data subsets may result in different detection generalization performances [10].

Meanwhile, cloud computing has been widely worked with existing detection schemes to discover attacks [11]. Nevertheless, due to the high latency and low scalability of the cloud, cloud-based detection schemes usually cost a higher detection time to identify attacks [2]. This indicates the cloud-based schemes may not be directly applied in real-time healthcare environment. To solve this issue, one of the promising solutions is to implement edge nodes by placing computing resources and tasks closer to the edge of the network [12]. However, edge computing still faces potential threats. For example, in an edge-based remote patient monitoring environment, an attacker may compromise edge nodes by easily launching different attacks, such as distributed denial of service (DDoS). This may increase the malicious attackers' capabilities. Hence, the aforementioned challenges should be taken into account when developing an attack detection scheme in e-healthcare system.

In this paper, we propose an Edge-assisted Anomaly Detection (EAD) scheme to detect attacks according to their abnormal behaviors in e-healthcare system. Since irrelevant and redundant features may degrade the detection accuracy,

978-1-7281-8104-2/21/$31.00 ©2021 IEEE

we investigate the feature selection to assist in discriminating normal behaviors and malicious attacks. Then, we exploit two different ensembles, i.e., bagging and random subspace, to enhance the detection rate and stability. Bagging means that samples are randomly drawn with replacement. Random subspace performs in the feature space by randomly selecting a subset of features. By combing them from example space and feature space, we construct a composite ensemble method better than single ensemble. To reduce the high latency caused by the cloud, we utilize edge nodes to perform EAD scheme locally. The main contributions of this paper are as follows.

1) To solve the issue of low detection accuracy caused by the irrelevant and redundant features of data, we propose a wrapper feature selection method. This method relies on binary particle swarm optimization, which selects an optimal subset of relevant and important features to make the detection scheme more efficient.
2) To effectively detect malicious attacks (e.g, DDos, Probe, and Worms), we propose an ensemble learning method by combining bagging and random subspace. It generates diverse classifiers and then selects a subset of effective classifiers by ensemble selection.
3) To reduce the latency of attack detection at the cloud, we propose an edge-assisted anomaly detection scheme. We evaluate the proposed scheme through extensive simulations on two well-known datasets. The proposed EAD scheme achieves a higher accuracy and detection rate compared with other state-of-the-art detection methods.

The remainder of this paper is organized as follows. In section II, we discuss the related works concerning attack detection. Section III introduces system model and attack model. Then, we present details of the proposed EAD scheme in Section IV. The performance evaluation is provided in Section V. Finally, the paper concludes in Section VI.

II. Related work

Anomaly detection schemes have received considerable attentions and extensive studies have been developed in recent years. Using statistical models, Moustafa et al. [3] proposed a novel outlier dirichlet mixture mechanism for detecting malicious activities in Internet of Things (IoT) networks. This anomaly detection mechanism can easily update the parameters of a profile for normal traffic, thereby improving the detection performance. However, it requires a large number of purely legitimate instances to ensure the best performance of the mechanism. Pajouh et al. [13] developed an intrusion detection model, which utilizes linear dimension analysis for dimension reduction and a mix of naive bayes and certainty factor version of k-nearest neighbor for classification. Then, they developed an improvement model (TDTC) and introduced principal component analysis in feature reduction module [7]. Although this model relatively solves the problem of insufficient handling of rare attacks, it is still incapable of achieving a proper detection rate against routine and less dangerous attacks, such as DoS and Probe attacks. Kanakarajan et al. [6] proposed a new tree ensemble method, which applies greedy randomized adaptive search procedure with annealed randomness and uses information gain to improve detection accuracy. Nevertheless, they do not consider the quality of the generated tree classifier. Similarly, Pham et al. [10] utilized gain ratio technique as feature selection and bagging to combine tree-based classifiers. Their model achieved a significant result in detection performance (84.25%) with using J48 classifier and only 35 features. In [14], to identify malicious activities, Yang et al. proposed a fuzzy aggregation method combining the modified density peak clustering algorithm (MDPCA) and the deep belief networks (DBNs). The MDPCA and DBN are utilized to reduce the imbalance of multi-class network data and extract high-level abstract features from dataset. Despite a high accuracy rate in identifying normal behaviors, the method performs poorly in detecting low frequency and dangerous attacks. More importantly, it is unable to support real-time detection. In addition, Alzubi et al. [15] used modified grey wolf optimization algorithm and SVM classifier to detect anomaly patterns. This approach models malicious and normal behaviors but has a yet low detection accuracy.

To solve the limited detection capabilities of medical devices (e.g., low memory, energy consumption, and computation power), cloud/ edge-based detection schemes have been proposed in e-healthcare system. With using an ensemble of online sequential extreme learning machine, Alrashdi et al. [12] proposed a fog-based scheme for detecting malicious activities in e-healthcare environment. This study gives the importance of e-healthcare for monitoring patients base on fog computing. In [16], Diro et al. proposed a distributed detection scheme in fog computing based on deep learning to identify patterns of attacks. Each distributed fog node collects the traffic data for training and processing at the edge of the network, and then share and optimize in a coordinating node. However, the used stochastic gradient descent method may cause time complexity and error functions in neural network, so that it can be hardly applied in e-healthcare scenarios. Aiming at the security issues of sharing health data via cloudlet, Chen et al. [17] proposed a trust model by using encryption, which enables the identification of reliable destinations (e.g., hospitals, medical offices) to share data and helps to connect patients with doctors as well.

In summary, most existing studies focused on the design of cloud-based methods for attack detection. However, many anomaly detection schemes cannot be directly applied in e-healthcare system, due to the inability of supporting the unique requisites of the e-healthcare environment such as scalability and low latency. Moreover, these detection schemes still suffer low accuracy and detection rates to identify multiple attacks in environment containing a small amount of training data. To this end, we aim to reduce computational overhead, while improving detection accuracy and stability. Edge computing is applied to reduce the detection latency of the cloud.

III. System Model and Attack Model

In this section, we introduce the system model and attack model, including three entities and four types of attackers.

A. System Model

The general view of our system model for edge-assisted attack detection is sketched in Fig. 1. Our system has three entities, i.e., trusted authority, users, and edge nodes.

• **Trusted Authority (TA)** bootstraps the entire system and generates identities for users. In addition, TA reviews the user data stored in the edge nodes. After detecting malicious activities, the TA can generate alerts and notify the cloud.

• **Users** such as doctors, patients and their families, take mobile smartphones to receive sensing data from medical and wearable devices. They can also transmit these health data to edge nodes for further processing and analysis. In addition, they can be sensing objects (e.g., patients) of medical and wearable devices, from which health data can be collected.

• **Edge Nodes** located at the edge network have network resources, storage capability, and computing power (e.g., e-healthcare gateways, routers, and switches). Moreover, edge nodes can directly communicate with users and collect their data because they are in proximity to users.

B. Attack Model

According to the attacker's capabilities, we define four types of malicious attackers in e-healthcare system.

1) *Attackers Limiting User Activity* (Type-1): The attackers attempt to prevent user from reading/writing medical resources. Based on their knowledge or experience, they may utilize any vulnerabilities, e.g., buffer overflow. This process can crash medical services, making it impossible for users to operate them. These attackers generally contain Generic, Fuzzers, Exploits, and Worms.

2) *DDos/DoS Attackers* (Type-2): A DDos/DoS attacker sends flooding of superfluous requests to smartphones to interrupt the normal health services for users. The attacker exhausts the resources of the target through brutal means, so that wearable and medical devices cannot provide normal services or resource access. In addition, the attackers can deny legitimate users access to healthcare apps.

3) *Probe Attackers* (Type-3): Probe attackers aims to collect network information and scan the target wearable devices by sending probe packets. They may utilize the collected scanning information, such as the IP addresses of devices, to pretend to be normal users to evade system security controls. Thus, they may find out vulnerabilities and disrupt the system.

4) *Unauthorized Access* (Type-4): An attacker, who does not have an account, tries to gain unauthorized access to the target system. The attacker may utilize some means, such as sending packets and password guessing, to intrude into the victim user. Consequently, the attacker may destroy users' health information. Although the number of Type-4 attackers is relatively rare, these attackers are dangerous for users.

In e-healthcare environment, there exist a large amount of routine and low dangerous attacks, i.e., Type-1, Type-2, and Type-3 attackers. Although these attacks are common, it is possible to disrupt the e-healthcare system by using system vulnerabilities, making users incapable to access the corresponding medical services. On the other hand, Type-4 attackers can be dangerous compared to other types because they are low frequency and distribution for on-site sampling and analysis and can also cause serious damages.

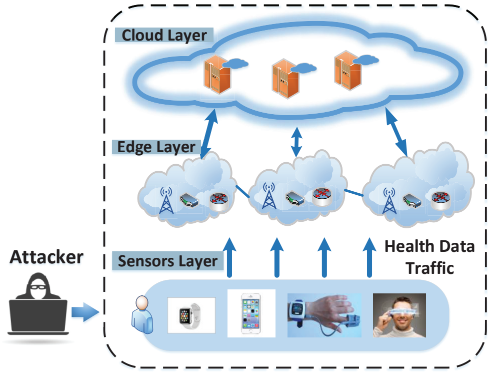


Fig. 1. System model

As shown in Fig. 1, in e-healthcare system, thousands of medical and wearable devices are located at the sensors layer to monitor users. Attack detection scheme is supported at the edge layer, where edge nodes are responsible for detecting attacks. Each fog node can be connected to specific sensing devices through the base stations, and detect attacks by processing health data traffic from network devices. Once detecting suspicious activities, TA generates alerts and notifies the remote cloud.

IV. The Proposed EAD Scheme

In this section, we describe the detailed process the EAD scheme. As shown in Fig. 2, the proposed EAD scheme contains data pre-processing and ensemble classification modules.

A. Data Pre-processing

To effectively detect attacks in e-healthcare system, data pre-processing module converts the raw data into clean data, which is utilized for training classifiers to achieve accurate and efficient results. It consists of the following three main steps.

Firstly, a feature conversion replaces symbolic features with numerical ones. This is because a well-trained classifier only accepts each feature record, which is represented as a real number vector. The converting process is considered essential and has an impact on detection accuracy. Thus, we handle the symbolic features such as replacing the values with ordered numbers, e.g., TCP = 1, UDP = 2, and ICMP = 3.

Then, a feature selection is the critical process of selecting relevant and important features. In other words, redundant and noisy features slow down the classification process, or even degrade the classifier's performance and training efficiency. More importantly, exploiting some relevant features helps the scheme better detect attacks. To this end, we propose a wrapper feature selection method based on binary particle swarm optimization (BPSO) [18]. Generally, wrapper-based methods consist of three components: feature search strategy, learning algorithm, and fitness function [19].

Without loss of generality, the feature selection is formulated as an optimization problem with the objective to

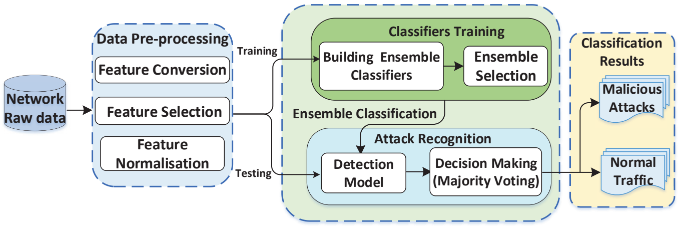


Fig. 2. The proposed EAD scheme.

minimize the fitness function. The fitness function should consider three objectives to assess a subset of selected features: detection rate (ϕ), false positive rate (ω), and the number of selected features (G). We have considered these three objectives simultaneously, because an individual objective cannot select effective features. For example, when more features are selected, the detection rate may be low, which may eventually lead to failure to detect attacks. Let the full feature set be F and G is a subset of the feature set. To obtain the optimal subset, we formulate an optimization problem in Eqn. (1).

$$\begin{aligned} \min_{G} \text{ Fitness} = \ & \alpha \cdot \frac{1}{\phi(G)} + \beta \cdot \omega(G) + \gamma \cdot \frac{|G|}{|F|}, \\ s.t. \quad & 0 < \phi(G) < 1, \\ & 0 < \omega(G) < 1, \\ & 0 < |G| < |F|, \ |G| = 1, 2, ..., N-1, \\ & \alpha + \beta + \gamma = 1, \end{aligned} \tag{1}$$

where $|G|$ is the number of selected features, $|F|$ is the total number of features, and α, β, and γ are the control parameters of the fitness function.

Since the objective optimization model is nondifferentiable, we utilize the heuristic BPSO as a feature search strategy to solve this problem. In BPSO method, a feature subset is represented by a particle in the swarm. Each particle position that denotes binary strings is evaluated by the fitness function, and the velocity exists in continuous space. Velocity is mapped to a scalar value between 0 and 1 using a sigmoid function. This scalar value is interpreted as the probability that the corresponding part of the binary position string is bit 1 or 0. The BPSO determines the global optimal solution by updating the flying velocity and the position of individuals in the group according to Eqn. (2).

$$v_{ij}^{t+1} = wv_{ij}^{t} + c_1 r_{1j}(\bar{x}_{ij}^{t} - x_{ij}^{t}) + c_2 r_{2j}(\hat{x}_{j}^{t} - x_{ij}^{t}). \tag{2}$$

where v_{ij}^{t+1} denotes the velocity of particle i in dimension j at iteration t, c_1 and c_2 are cognitive and social constants, $\bar{x}_{ij}^{t}$ and x_{ij}^{t} denote the best current position and the position of particle i in dimension j at iteration t, respectively. $\hat{x}_{j}^{t}$ represents the position of the whole population in dimension j, w is the inertia weight, and r_{1j} and r_{2j} are random values.

Then, an optimal feature subset is determined by considering the minimum fitness of a particular classifier. In this paper, we utilize one of the popular decision tree algorithms, called C4.5 classifier [20] as learning algorithm due to its simplicity and speed in generating trees.

Finally, a feature normalization is to scale the value of each feature into a certain range $[0, 1]$. This scaling helps to speed up gathering and removing the bias from the raw data without altering their statistical properties. A simple and fast method is *min-max* normalization [3]. At the end of these steps, the data is prepared for the classifier training and testing phases.

B. Ensemble Classification

After pre-processing, an optimal and important feature subset is passed to ensemble classification module to facilitate the recognition of attacks. The attackers in Section II B are very dangerous to the system. Type-1 and Type-2 attackers are common in reality. Type-1 attackers may find the vulnerabilities in the apps, such as buffer overflow, and exploit the user's data to enter the e-healthcare system. Type-3 attackers try to collect useful information about target users by ports scanning. Moreover, Type-4 attackers may try to gain unauthorized access to the system like brute force password guessing. In this module, we focus on detecting the four types of attackers by exploiting ensemble learning. The idea of ensemble learning is to combine a set of individual weak classifiers in a certain way, and then average the output of multiple classifiers to achieve prediction results with higher stability and accuracy [21]. Even if a weak classifier gets a wrong classification prediction, other weak classifiers can also correct the error. Therefore, these weak classifiers can produce a powerful model to complete attack detection tasks. To this end, we exploit bagging and random subspace methods to construct an ensemble of diverse and robust classifiers to detect these attacks.

Bagging is based on the concept of bootstrap aggregating [22], which randomly selects samples with replacement. Given a training set D^{tr} and a classifier C_i, bagging randomly generates new training sets with replacement. From each bootstrap training set D_i $(i = 1, ..., M)$, a base classifier C_i is induced by the same learning algorithm. By voting the predictions of each of these classifiers, bagging can seek to reduce the error due to variance of the base classifier. The random subspace (RSS) is also an ensemble construction technique proposed by Ho et al. [23]. It randomly selects a certain proportion of $\mu * d$ $(0 < \mu \leq 1)$ dimensional features from the original d dimensional feature set. One may obtain better classifiers in random subspaces than in the original feature space.

By utilizing the characteristic of the two methods, the training dataset is modified in two ways (e.g., horizontal and vertical). Specifically, each individual classifier is built by drawing random subsets of both samples (horizontal) and features (vertical). First, let $D^{tr} = (\mathbb{X}^1, \mathbb{X}^2, ..., \mathbb{X}^N)$ be the training dataset and its size is N. The bagging produces a bootstrap sample $D_i = (\mathbb{X}_i^1, \mathbb{X}_i^2, ..., \mathbb{X}_i^N)$, which is randomly drawn (with replacement) from D^{tr} $(i = 1, ..., M$, where M is the size of bag). Then, we randomly select p $(p = \mu * d, 0 < \mu \leq 1)$ features from each bootstrap sample. Thus, the p dimensional random subspace of the original d dimensional feature space is obtained. The independent partition of feature set can generate a variety of the selected features. Note that

Algorithm 1 Ensemble Learning Method

```
Input:
    The training set $D^{tr}$, the validation set $D^{va}$, The size of bag $M$, The dimension proportion of feature set $\mu$, The proportion of selected classifiers, $k$;
Output: Ensemble of classifiers $\widetilde{E}$;
Initialize $E = \varnothing$;
for each $i \in [1, M]$ do
    $D_i$ = Bootstrap sample from training set $D^{tr}$ (i.d.d. sample with replacement);
    $\widetilde{D}_i$ = Select Random Subspace from $(D^{tr}, \mu * d)$;
    $C_i$ = Build base classifier on $\widetilde{D}_i$;
    $E = E \bigcup \{C_i\}$;
end for
Ensemble selection based on base classifiers' performance on validation set $D^{va}$;
$\widetilde{E}$ = Select $k * M$ an optimal subset of the classifiers from $E$;
return $\widetilde{E}$;
```

the new training set $\widetilde{D}_i = (\widetilde{\mathbb{X}}_i^1, \widetilde{\mathbb{X}}_i^2, ..., \widetilde{\mathbb{X}}_i^N)$ consists of p dimensional training samples $\widetilde{\mathbb{X}}_i^j = (x_i^{j1}, x_i^{j2}, ..., x_i^{jp})$, where the p components x_i^{jk} $(k = 1, 2, ..., p)$ are randomly selected from d components. Then, a base-level classifier C_i of the random subspace $\widetilde{D}_i$, $i = 1, 2, ...M$, is constructed. Some of these M base classifiers have good detection accuracy, while others may have mediocre or even poor accuracy.

Simply combining good and bad classifiers may degrade the accuracy of the entire scheme. To solve this problem, we adopt static ensemble selection [24] from the ensemble pool to find a subset of classifiers that produces excellent performance when averaged together. The goal of ensemble selection is to select a subset of classifiers $\widetilde{E}$, where $\widetilde{E} \subset E$, which has the most effective classifiers on the validation set, where test samples are classified based on all local criteria. More specifically, we apply the heuristic method to select $k * M$ $(0 < k < 1)$ most effective classifiers. It needs to sort the classifiers according to their performance, and then determines the number of classifiers that consist of the optimal set. The Classification And Regression Tree classifier (CART, a type of decision tree algorithms) [20] is considered as base-level algorithm to select effective classifiers. The last step is the majority voting in testing phase, which means that the predictions of the selected classifiers are combined to make final decisions. The process of ensemble learning method is shown in Algorithm 1.

V. Performance Evaluation

A. Simulation Settings

Due to popularization, privacy, and commercialization issues, there is a lack of public e-healthcare benchmark datasets. The attacks launched on conventional computer networks can be used to simulate the attacks conducted in e-healthcare system. To mimic four types of attackers, we select NSL-KDD [25] and UNSW-NB15 [26] as the benchmark datasets to assess the proposed EAD scheme. Each dataset is similar to the real-world's attack situation. The NSL-KDD dataset contains a large number of Type-2, Type-3, and Type-4 attacks, accounting for 99.91% of all attacks. In the UNSW-NB15 dataset, 85.63% of attacks are Type-1 and Type-3.

TABLE I
Performance results for detection scheme based on NSL-KDD and UNSW-NB15 datasets when using BPSO feature selection.

Datasets	Features	Evaluation metric			
		Accuracy(%)	**DR(%)**	**Training(s)**	**Testing(s)**
NSL-KDD	All	80.53	69.18	12.72	0.83
	Selected	90.06	85.42	3.87	0.36
UNSW-NB15	All	86.86	96.09	26.73	0.48
	Selected	90.59	96.43	3.47	0.27

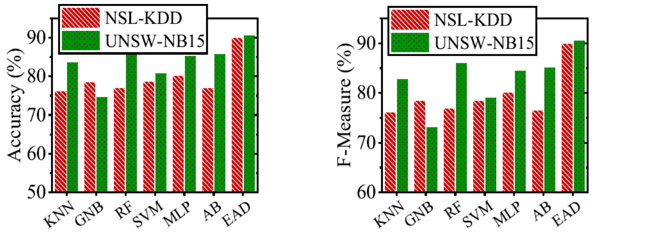
The evaluation metrics include Accuracy, Detection Rate (DR, as known as Recall), False Positive Rate (FPR), F-Measure, and Detection Time, which are explained in [16]. For NSL-KDD dataset, we use KDDTrain_20%+ as training set (25,192 records, including 13,449 normal and 11,743 attacks) and KDDTest+ as testing set (22,544 records, including 9,711 normal and 12,883 attacks). For UNSW-NB15 dataset, we apply the stratified sampling to randomly select 20% of records from UNSW_NB15_training-set and UNSW_NB15_testing-set as our training (35,069 records, including 11,200 normal and 23,869 attacks) and testing set (16,466 records, including 7,400 normal and 9,066 attacks). The experiments are conducted using independent 10 runs. For each run, the original training set is randomly divided on the basis of 75% for training and the remaining 25% as a validation set. For proposed ensemble method, we set $M = 200$ and $k = 0.5$. Through the experiments, we set $\mu_{nsl} = 1.0$ and $\mu_{unsw} = 0.8$ for two datasets. In BPSO, the size of the population, maximum number of iterations, inertia weight, and position constant are set to 30, 100, 0.9, 2, respectively [18]. In addition, $\gamma = 0.1$ and $\alpha = \beta = 0.45$ since DR and FPR are considered as equally important. The proposed EAD scheme is implemented using Python with libraries (e.g., Scikit-Learn) on 64-bit OS, equipped with Intel i7 CPU and 16 GB RAM.

B. Results Analysis

Results of BPSO and ensemble methods. We employ the BPSO method to select important features. For NSL-KDD dataset, 15 features are chosen from original 41 features, that is f_1,f_2,f_5,f_6,f_9,f_{13},f_{16},f_{21},f_{22},f_{25},f_{28},f_{38}. For UNSW-NB15 dataset, 9 features are selected from original 42 features, that is f_3,f_4,f_7,f_{11},f_{20},f_{28},f_{30},f_{32},f_{37}. Table I shows the detection scheme with selected features in all aspects significantly outperforms that of the detection scheme using all the features. From table II, it can be seen that after feature selection, the proposed ensemble method performs better than all other classifiers on both datasets, and has a smaller standard deviation. This proves the effectiveness of the proposed ensemble method. Fig. 5 shows the accuracy of proposed ensemble method for the proportion of selected classifiers k value. It is noted that when the k value increases, the accuracy varies significantly. The results indicate that the k value will influence the accuracy and we need to set an appropriate k between 0.3 and 0.6 to identify attacks. If these attacks cannot be accurately detected, this will not only degrade the system performance and users' experiences but

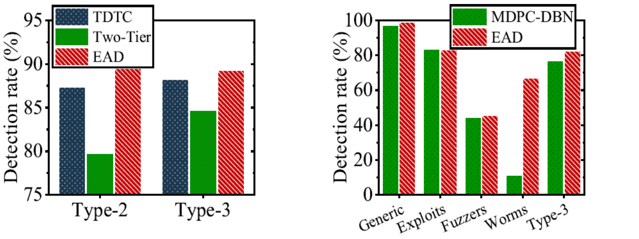
TABLE II
PERFORMANCE CLASSIFICATION BASED ON NSL-KDD AND UNSW-NB15 DATASETS WITH STANDARD DEVIATIONS.

Classifier	NSL-KDD				UNSW-NB15			
	Accuracy(%)	**DR(%)**	**FPR(%)**	**F-Measure(%)**	**Accuracy(%)**	**DR(%)**	**FPR(%)**	**F-Measure(%)**
Bagging	88.91 ± 1.45	83.53 ± 2.49	3.96 ± 0.28	88.87 ± 1.44	88.08 ± 0.03	95.29 ± 0.06	19.13 ± 0.08	88.15 ± 0.04
RSS	88.74 ± 1.10	83.54 ± 1.94	4.37 ± 0.01	88.70 ± 1.09	88.23 ± 0.65	96.26 ± 0.29	21.61 ± 1.81	87.84 ± 0.72
BRS	89.33 ± 0.73	84.13 ± 1.34	3.81 ± 0.17	89.28 ± 0.72	89.98 ± 0.64	96.53 ± 0.36	19.61 ± 1.88	88.95 ± 0.72
Proposed Ensemble	90.06 ± 0.52	85.42 ± 0.95	3.80 ± 0.15	90.01 ± 0.51	90.59 ± 0.22	96.43 ± 0.12	18.70 ± 0.21	90.63 ± 0.12



(a) Accuracy vs. machine learning (b) F-Measure vs. machine learning

Fig. 3. Performance comparison of the proposed EAD for binary classification with different machine learning on NSL-KDD and UNSW-NB15 datasets.



(a) Detection rate vs. attacks in NSL-KDD dataset (b) Detection rate vs. attacks in UNSW-NB15 dataset

Fig. 4. Detection rate for different attacks in two datasets.

also destroy e-healthcare system.

To validate the proposed EAD scheme, we compare proposed EAD with six well-known machine learning methods, namely, K-Nearest Neighbor (KNN), Gaussian Naive Bayes (GNB), Random Forest (RF), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and AdaBoost (AB). Fig. 3 illustrates our proposed EAD can achieve higher accuracy and F-measure than other methods on two datasets. We also compare EAD performance with other existing anomaly detection methods, namely, GAR-Forest [6], Two-Tier [13], TDTC [7], MBGWO [15], BREPtree [10], EM Clustering [26], Neural Network [26], and MDPCA-DBN [14], which are explained in Section II. The Table III is binary classification (normal and attacks). The EAD can perform better than the above methods as it precisely considers the quality and diversity of classifiers. In Fig. 4 (a), compared with TDTC and Two-tier methods, EAD can detect Type-2 and Type-3 attacks more effectively. In addition, the detection rate for Type-4 attackers is 62.42%, which is slightly low. This is because the behaviors of Type-4 attackers are very similar to normal records, making it difficult for various methods to distinguish them. As can be seen in Fig. 4 (b), the EAD is more effective than MDPC-DBN in detecting Generic, Exploits, Probe, and Worms. Furthermore, the runtime complexity of the EAD is analyzed and compared with those of other state-of-the-art methods. Assuming that N, f, t, and l are the number of instances, the number of features, the number of trees in ensemble model, and the number of heuristic iterations in BPSO, respectively. The computational complexity of the BPSO feature selection is $O(Nfl)$. For the ensemble method, its computational complexity is $O(Nft+N(\log_2 f)t) \approx O(Nft)$. Thus, the overall computational complexity of the EAD is $O(Nfl+Nft)$. For other methods, k is the number of clusters, m is the number of iterations, and n is the number of neurons. Obviously, the MDPC-DBN has the largest runtime complexity, since it needs pre-training and fine-tuning. As the number of samples and features increases, EAD runs more quickly than the others. The existing methods either do not consider the quality of the classifiers or are too computationally expensive. In summary, the proposed EAD can achieve a higher detection rate with low computational overhead. In other words, the EAD can detect these attacks efficiently, which timely protects users from being endangered in e-healthcare system.

TABLE III
COMPARISON OF EAD WITH OTHER DETECTION METHODS (N/A MEANS NO AVAILABLE RESULTS, AND ACC MEANS ACCURACY).

Method	**Dataset**	**ACC(%)**	**DR(%)**	**FPR(%)**	**Complexity**
GAR-Forest [6]	NSL-KDD	85.05	85.1	12.2	$O(Nf + Nftl)$
Two-Tier [13]	NSL-KDD	N/A	83.24	4.8	$O(Nf^2 + 2Nf)$
TDTC [7]	NSL-KDD	N/A	84.82	5.56	$O(2Nf^2 + 2Nf)$
BREPtree [10]	NSL-KDD	83.22	N/A	8.09	$O(Nf + Nft)$
MBGWO [15]	NSL-KDD	81.58	N/A	N/A	$O(N^2fl)$
proposed EAD	NSL-KDD	90.06	85.42	3.8	$O(Nfl + Nft)$
EM Clustering [26]	UNSW-NB15	78.47	N/A	23.79	$O(kNf)$
Neural Network [26]	UNSW-NB15	81.34	N/A	21.13	$O(Nfmn^2)$
MDPC-DBN [14]	UNSW-NB15	90.21	96.22	17.15	$O(kNf+Nfmn^2)$
Proposed EAD	UNSW-NB15	90.59	96.43	18.70	$O(Nfl + Nft)$

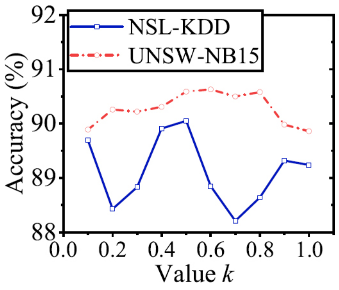


Fig. 5. The accuracy on two datasets with k values.

Results of detection time. Finally, we evaluate the detection time by varying the total amount of data traffic between edge-

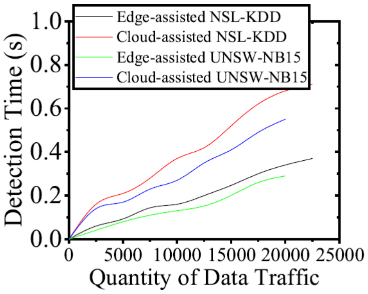


Fig. 6. Detection time comparison of edge-assisted and cloud-assisted schemes.

based and cloud-based schemes. In Fig. 6, as the testing network traffic grows, the detection time of the two schemes also increases. However, the proposed edge-assisted scheme always achieves a lower detection time than the cloud-assisted scheme since the edge nodes are closer to wearable devices. This suggests that edge-assisted detection is an effective way in e-healthcare system, where real-time detection is required.

VI. Conclusion

In this paper, we have proposed an edge-assisted anomaly detection scheme based on ensemble learning to detect malicious attacks in e-healthcare system. We have investigated the BPSO feature selection method to select an optimal feature subset to enhance detection accuracy. The proposed ensemble learning method can also detect the four types of attackers. The simulation results show that the EAD scheme can achieve high detection accuracy with reasonable overhead. Furthermore, the proposed EAD scheme is a new paradigm in e-healthcare system, taking the advantages of low latency and high detection capabilities in the edge nodes, while protecting users' health data. In the future, we will investigate online learning method to enhance the ability of the detection scheme.

Acknowledgment

This work has been supported by a research grant from the National Key Research and Development Program of China (No. 2019JSJ12ZDYF01), and the Fundamental Research Funds for the Central Universities with grant number 2020GFZD014.

References

[1] Forbes, available:http://www.forbes.com/.

[2] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 1, pp. 601–628, 2018.

[3] N. Moustafa, K. Choo, I. Radwan, and S. Camtepe, "Outlier dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 8, pp. 1975–1987, 2019.

[4] M. Han, B. Kwak, and H. Kim, "Event-triggered interval-based anomaly detection and attack identification methods for an in-vehicle network," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2941–2956, 2021.

[5] A. Hamamoto, L. Carvalho, L. Sampaio, T. Abrão, and M. Proença, "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Syst. Appl.*, vol. 92, pp. 390 – 402, 2018.

[6] N. Kanakarajan and K. Muniasamy, "Improving the accuracy of intrusion detection using gar-forest with feature selection," in *Prof. of FICTA*, 2015, pp. 539–547.

[7] H. Pajouh, Reza, R. Khayami, A. Dehghantanha, and K. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks," *IEEE Trans. Emerg. Top. Comput.*, vol. 7, no. 2, pp. 314–323, 2019.

[8] S. Garg, K. Kaur, S. Batra, G. Kaddoum, N. Kumar, and A. Boukerche, "A multi-stage anomaly detection scheme for augmenting the security in iot-enabled applications," *Futur. Gener. Comput. Syst.*, vol. 104, pp. 105–118, 2020.

[9] L. Zhou, J. Shu, and X. Jia, "Collaborative anomaly detection in distributed SDN," in *Proc. of IEEE GLOBECOM*, 2020, pp. 1–6.

[10] N. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection," in *Proc. of ACSW*, 2018, pp. 1–6.

[11] B. Zarpelão, R. Miani, C. Kawakani, and S. Alvarenga, "A survey of intrusion detection in internet of things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, 2017.

[12] I. Alrashdi, A. Alqazzaz, R. Alharthi, E. Aloufi, M. Zohdy, and H. Ming, "FBAD: fog-based attack detection for iot healthcare in smart cities," in *Prof. of IEEE UEMCON*, 2019, pp. 515–522.

[13] H. Pajouh, G. Dastghaibyfard, and S. Hashemi, "Two-tier network anomaly detection model: a machine learning approach," *J. Intell. Inf. Syst.*, vol. 48, no. 1, pp. 61–74, 2017.

[14] Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, "Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks," *Appl. Sci.*, vol. 9, no. 2, pp. 1–25, 2019.

[15] Q. M. Alzubi, M. Anbar, Z. N. M. Alqattan, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on a modified binary grey wolf optimisation," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 6125–6137, 2020.

[16] A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," *Futur. Gener. Comput. Syst.*, vol. 82, pp. 761–768, 2018.

[17] M. Chen, Y. Qian, J. Chen, K. Hwang, S. Mao, and L. Hu, "Privacy protection and intrusion avoidance for cloudlet-based medical data sharing," *IEEE Trans. Cloud Comput.*, vol. 8, no. 4, pp. 1274–1283, 2020.

[18] K. Chen, F. Zhou, and X. Yuan, "Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection," *Expert Syst. Appl.*, vol. 128, pp. 140–156, 2019.

[19] A. Nazir and R. A. Khan, "A novel combinatorial optimization based feature selection method for network intrusion detection," *Comput. Secur.*, vol. 102, pp. 1–12, 2021.

[20] K. Rai, M. Devi, and A. Guleria, "Decision tree based algorithm for intrusion detection," *Int. J. Adv. Netw. Appl.*, vol. 07, pp. 2828–2834, 2016.

[21] F. Salo, A. Nassif, and A. Essex, "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection," *Comput. Networks*, vol. 148, pp. 164–175, 2019.

[22] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, no. 1-2, pp. 1–39, 2010.

[23] T. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.

[24] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Inf. Fusion*, vol. 6, no. 1, pp. 63–81, 2005.

[25] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Prof. of CISDA*, 2009, pp. 1–6.

[26] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J. A Glob. Perspect.*, vol. 25, no. 1-3, pp. 18–31, 2016.