Journal of Computer Science and Cybernetics, V.39, N.1 (2023), 1–16
DOI no 10.15625/1813-9663/16675

ONE-CLASS FUSION-BASED LEARNING MODEL FOR ANOMALY DETECTION

CONG THANH BUI[1], VAN LOI CAO[2], MINH HOANG[3], QUANG UY NGUYEN[2,*]

[1]*Software R&D Department, Hi-tech Telecommunication Center, Communications Command, Ha Noi, Viet Nam*

[2]*Faculty of Information Technology, Le Quy Don Technical University, 236 Hoang Quoc Viet Street, Bac Tu Liem District, Ha Noi, Viet Nam*

[3]*Vietnam Institute of Science Technology and Innovation, 113 Tran Duy Hung Street, Cau Giay District, Ha Noi, Viet Nam*

Abstract. The Dempster-Shafer (DS) theory of evidence is frequently used to combine multiple supervised machine learning models into a robust fusion-based model. However, using the DS theory to create a fusion model from multiple one-class classifications (OCCs) for network anomaly detection is a challenging task. First, the lack of attack data leads to the difficulty in estimating an appropriate threshold for the OCC models to distinguish between normal and abnormal samples. Second, it is also very challenging to find the weight of OCCs that corresponds to the contribution of each OCC model in the fusion model. In this paper, we attempt to solve the above issues in order to make the DS theory applicable to constructing OCC-based fusion models. Specifically, we propose two novel methods for automatically choosing the appropriate threshold of OCCs and for estimating the weight of individual OCCs in fusion-based models. Thanks to that, we develop a One-class Fusion-based Anomaly Detection model (OFuseAD) from multiple single OCCs. The proposed model is evaluated on ten well-known network anomaly detection problems. The experimental results show that the performance of OFuseAD is improved on almost tested datasets using two metrics: accuracy and F1-score. The visualization results provide insight into the characteristics of OFuseAD.

Keywords. Deep learning, AutoEncoder, anomaly detection, DS theory, data fusion.

1. INTRODUCTION

Developing security systems has received great attention from both research and industrial communities. Mining meta-data (i.e, network traffic and log files) to identify security events is one of the primary tasks for almost every security system including Intrusion Detection Systems (IDSs) and network monitoring systems. In IDSs, the detection engines are often constructed based on two approaches: signature-based and anomaly detection-based techniques. The former works by matching malicious signatures with patterns found in data using rules. This approach is highly accurate on known attacks, but it is unable to identify

*Corresponding author.
E-mail addresses: congthanhttmt@gmail.com (C.T. Bui); loi.cao@lqdtu.edu.vn (V.L.Cao); hoangminh@most.gov.vn (M. Hoang); quanguyhn@gmail.com (Q.U. Nguyen).

new/unknown ones. On the contrary, the latter can learn to capture the normal behaviors of network traffic data. The resulting model is then used to assign anomalous scores (or labels) for given a querying data point. This approach is very flexible and capable of detecting novelty network attacks.

Anomaly detection techniques have been widely developed for identifying security events from computer and IoT networks in recent years [1–4]. One-class classification (OCC) is known as a typical approach for network anomaly detection. The OCCs exhibit several advantages such as requiring only normal data in the training phase, the ability to handle high-dimensional network data, and enhancing the ability in detecting new/unknown malicious activities [5–7]. Traditional OCC-based techniques can be categorized into two main groups: density-based methods and distance-based methods [8]. Recently, deep learning, such as deep AutoEncoders (AEs), has emerged as the state-of-the-art anomaly detection method [5, 6]. However, each kind of OCC-based method including AEs often performs well only on some types of problems [3, 9–11]. Therefore, aggregating different kinds of OCC-based learners to construct a fusion anomaly detection model may potentially further improve the performance of single OCCs.

In the field of fusion learning, the Dempster-Shafer (DS) theory of evidence is a method for establishing fusion-based anomaly detection models [9, 12–15]. This method allows to combine the evidences supporting a hypothesis (normal or anomaly) created by many source classifiers. The beliefs (i.e., anomalous scores) produced by individual classifiers for a given hypothesis can be fused using the DS rules to provide a unified view of the system state. The advantage of the DS theory is that no prior knowledge is required, making it potentially suitable for classifying previously unseen information [16].

However, applying the DS theory requires computing the probability supporting a given hypothesis (called mass value), such as normal, anomaly, or uncertainty, from each of the involved classifiers. In addition, to construct a good fusion-based model, it is important to weigh the involved classifiers according to their performance [9, 15, 17]. Therefore, it is very challenging to apply the DS theory to one-class classification due to the lack of abnormal data to validate the performance of the one-class models [11].

In this paper, we propose a one-class fusion-based anomaly detection model (OFuseAD) that overcomes the above issues. To obtain that, a method for automatically estimating the appropriate threshold for OCCs that allows OCCs can assign the normal and abnormal score for each data sample. We also introduce a novel metric for measuring the generalization ability of OCCs. This metric is then used to estimate the weight of OCCs in the fusion model. Finally, the DS theory is applied to create a one-class fusion-based model from multiple single OCCs. To the best of our knowledge, our research was the first attempt to address the two above difficulties when applying the DS theory to construct a fusion model from OCCs for network anomaly detection. Note that the contribution of this study focuses on making the DS theory more practical for fusing one-class-based anomaly detection models rather than mathematical developments on the DS theory.

The remaining of this paper proceeds as follows. In Section 3, we briefly review some relevant work. In Section 2, we present a complementary background of the paper. We detail our proposed model in Section 4. The experimental settings are described in Section 5. The results are highlighted and discussed in Section 6. Finally, the conclusion is presented in Section 7, which also draws some future directions.

2. BACKGROUND

This section presents the background of our proposed model. The presented methods include the DS theory, density-based anomaly detection, distance-based anomaly detection, and double-shrink auto-encoder.

2.1. The Dempster-Shafer theory of evidence

The Dempster-Shafer theory of evidence is proposed by Arthur Dempster and improved by Glenn Shafer [18]. The theory is used to calculate the probability of an event by combining the evidence from multi-sources. The proposition can be a subset of the given set of finite hypotheses, named Frame of Discernmen (FoD), and presented by Θ, which is known as the fusion problem under consideration having p exclusive and exhaustive hypothesis

$$\Theta = \{H_1, H_2, ...H_p\}. \tag{1}$$

Let $E_1, E_2, ..., E_n$ $(n \geq 2)$ be several evidence sources. The set of all subsets of Θ is denoted by 2^{Θ}. Basic Probability Assignment (BPA) over Θ is a function m: $2^{\Theta} \rightarrow [0, 1]$ that satisfies

$$\sum \{m(H))|H \subseteq \Theta\} = 1,\ m(\emptyset)) = 0, \tag{2}$$

where $m(H)$ represents the belief exactly committed to hypothesis H. In the DS theory, Dempster-Shafer's rule of combination (DRC) is used to combine mass assignments from multiple sources. When applying DRC to combine two sources E_i and E_j, the combined mass $m(H)$ obtained by combining E_i and E_j is detailed as follows

$$m(H) = \frac{\sum_{(B \cap C = H; B, C \subseteq \Theta)} \left[m_i(B)\, m_j(C)\right]}{1 - \sum_{(B \cap C = \varnothing; B, C \subseteq \Theta)} \left[m_i(B)\, m_j(C)\right]}, \tag{3}$$

where B and C are the variables that represent the hypotheses supported by the source E_i and E_j, respectively.

The DS combination rule in case of combining from many sources $E_1, E_2, ..., E_m$ can be done sequentially instead of globally [19]. To calculate the aggregation mass, $Mass(H)$, from three sources, the following equation provides the same result for any hypothesis $H \subseteq 2^{\Theta}$,

$$\begin{aligned} Mass(H) &= (m_1 \oplus m_2 \oplus m_3)(H) = ((m_1 \oplus m_2) \oplus m_3)(H) \\ &= ((m_1 \oplus m_3) \oplus m_2)(H) = ((m_2 \oplus m_3) \oplus m_1)(H). \end{aligned} \tag{4}$$

2.2. Density-based anomaly detection

Density-based methods estimate the density distribution of the input space and then identify anomaly objects as those lying in regions of low density. Let $X = \{x_1, x_2, ..., x_n\}$ be a set of examples in R^d drawn from an unknown distribution with a true density function $p(x)$. An estimation $\hat{p}(x)$ of the density function at sample x can be calculated using the equation as follows

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i), \tag{4}$$

where $K_h : x \rightarrow R$ is a kernel function with a smoothing parameter h called the bandwidth.

The density-based method used in this paper is Kernel Density Estimation (KDE) [20]. KDE fits a large number of kernels over the data, one per datapoint. These kernels, typically Gaussian kernel, share a single co-variance parameter, called bandwidth. The global probability density function is estimated by summing over the contributions of these kernels in which the probability density at a location depends on the datapoints lying within the localized neighborhood indicated by bandwidth. When applying KDE to network anomaly detection [5, 21], only normal data is used to construct a probability density function. A query datapoint will be classified as an anomaly if its density generated from the function is lower than a predetermined threshold. The main drawback of KDE is that it is very computationally expensive when working with large datasets.

2.3. Distance-based anomaly detection

Distance-based methods for anomaly detection are based on the calculation of distances among objects in the data. In distance-based approaches, nearest-neighbor methods are the most widely used to construct models for network anomaly detection. These methods assume that normal datapoints are located close to their neighbors in the normal training data, whereas anomalies occur far from those in the normal training data [8]. In this paper, the distance-based approach implemented is Local Outlier Factor (LOF) [22]. LOF views the datapoints that have a considerably lower local density than their neighbors as anomalies. It estimates a density deviation score, called the LOF score. The larger the score a given data point has, the higher the probability the data point is anomalous. LOF has shown its power in network anomaly detection, however, it has some limitations when dealing with high-dimensional data [5].

2.4. Double-shrink AutoEncoder

Shrink AutoEncoder (SAE) is an extension of AutoEncoders that learns a representation (encoding) for a set of data, usually for dimensional reduction [5, 21]. Double-shrink AutoEncoder (DSAE) [23] works based on incorporating one more shrink pressure into SAE. DSAE is trained on only normal data. The normal data is passed as the first input to DSAE. The output of the decoder will then also be used as the second input of DSAE. The model learns to minimize the reconstruction losses (call RE_1 and RE_2) and drive latent vectors (z^1 and z^2) toward the center of the latent space.

The first and the second reconstruction errors (RE_1 and RE_2) are denoted by the equations

$$L_{RE_1}(\theta, x_i) = \frac{1}{m}\sum_{i=1}^{m}\left(x_i - x_i^{out1}\right)^2, \tag{5}$$

$$L_{RE_2}(\theta, x_i) = \frac{1}{m}\sum_{i=1}^{m}\left(x_i^{out1} - x_i^{out2}\right)^2, \tag{6}$$

where θ is the parameters of DSAE, m is the number of the training data points, and x_i, x_i^{out1}, and x_i^{out2} are the input data point i and its reconstruction values in the first and second shrinks, respectively. The reconstruction error of the DSAE is the sum of the first and the second reconstruction error as in Eq. (7)

$$L_{RE}(\theta, x_i) = L_{RE_1}(\theta, x_i) + L_{RE_2}(\theta, x_i). \tag{7}$$

The shrink term of the DSAE loss is the sum of the first shrink and the second shrink components as presented in Eq. (8), (9) and (10)

$$L_{Z^1}(\theta, x_i) = \frac{1}{m}\sum_{i=1}^{m}\left\|z_i^1\right\|^2, \tag{8}$$

$$L_{Z^2}(\theta, x_i) = \frac{1}{m}\sum_{i=1}^{m}\left\|z_i^2\right\|^2, \tag{9}$$

$$L_Z(\theta, x_i) = L_{Z^1}(\theta, x_i) + L_{Z^2}(\theta, x_i). \tag{10}$$

When applying DSAE to anomaly detection, the distance of the representation of a data sample in Z^1 and Z^2 to the origin can be used as an anomaly score [23]. DSAE shows a good prediction ability on a wide range of well-known datasets.

3. RELATED WORK

This section briefly presents some studies that employ fusion techniques to construct network anomaly detection models. For a comprehensive survey of developing data fusion techniques for network intrusion detection, the readers are recommended to see Li et al. [12].

In the early days, Tim Bass et al. [24] proposed a fusion-based framework to improve intrusion detection systems. The authors provided general step by step to implement the architecture requirements using a multi-sensor data fusion. Giacinto et al. [25] proposed a fusion-based intrusion detection from several simple classifiers. Every single classifier is trained in a distinct subset of features, and the individual decisions are combined using several fixed and trainable fusion rules. Wang et al. [26] proposed a model using the DS theory of evidence to fuse knowledge from both network and host agents. Sisters and Maglaris [27] used the DS theory as the mathematical foundation for the development of a novel DoS detection engine. The detection engine is evaluated using real network traffic and shows a significant result. The DS theory was also used in the work of Hu et al. [28] where the authors attempted to solve the problem of how to analyze the uncertainty quantitatively. In their evaluation, the ingoing and outgoing traffic ratio, and service rate are selected as the detection metrics, and the prior knowledge in the DDoS domain is used to construct the basic probability assignment function (BPA).

Thomas and Balakrishnan et al. [29] improved the performance of IDS using the DS theory by fusing multiple differential IDSs together. They focused on choosing the detectors for fusion in order to get better advantages. The model was evaluated on the DARPA 1999 dataset and produced a potential result. Zhao et al. [30] used the DS theory to combine multiple anomaly detection methods into a more efficient fusion-based model. Their work used six classification algorithms to construct a fusion model. The results showed that choosing joined basic algorithms with the same reliability will lead the model to provide higher results. Liu et al. [9] proposed a new way of optimizing the DS theory to fuse six sensors. The authors introduced weights to control the balance among the anomaly detection algorithms. They also conducted the BPA function based on assuming that the distance among the normal records is less than that in the abnormal data. The model was evaluated

on the KDD-Cup99 dataset, with some anomalous labels for training the joined models. The authors claimed that their model provides better accuracy than any single one.

Recently, Chulin Lu et al. [31] proposed a hybrid Network Intrusion Detection System (NIDS) model using a neural network and the DS theory. The network traffic after preprocessing will be fed to an extractor model to divide into three subsets of features. Three supervised algorithms trained on these subsets of features are fused by the DS unit. Their model was then evaluated on the KDD-Cup99 dataset and is reported effectively. Shah et al. [15] introduced a modified framework of the DS theory when applying it to constructing a fusion-based model from multiple intrusion detection models. The model fused four detectors (two anomaly-based methods and two signature-based methods). The fusion model was evaluated on the KDD-Cup99 dataset and showed a significant result. Mattar et al. [17] introduced a network anomaly detection model using the DS theory. They proposed several ways to define a BPA function to fuse four different methods.

4. PROPOSED MODE

This section proposes the solutions for the two problems in Section 1 when applying the DS theory to construct a fusion model based on OCCs. The following is the list of our proposed solutions in this paper:

- A method for automatically choosing an appropriate threshold for OCCs.
- A new metric to measure the generalization ability of OCCs without anomalous data.
- A formulation for calculating the weight of each OCC in the fusion model.
- A fusion model for network anomaly detection based on the DS theory and OCCs.

4.1. Determining the threshold for OCCs

When applying OCCs to detect network anomalies, it is requested that an appropriate threshold is pre-set to separate normal and abnormal data. In this section, we propose a new method for automatically estimating the threshold for OCCs. The validation set, va, is used for determining such a threshold. We assume that normal data may contain a small percentage ($d\%$) of anomalies (also called contamination rate), then the threshold will be estimated to separate the normal and contaminated samples in va. Let f_j be the j-th OCC in OFuseAD and S^{va} be the set of anomaly scores of samples in va returned from f_j and sorted in the accent order. Let S^{va}_{sub} be a subset of S^{va} that contains the values from $0.90 * len(S^{va})$ to $0.98 * len(S^{va})$ [1], then the threshold is defined based on two consecutive values in S^{va}_{sub} that have the largest difference. In other words, if two consecutive values in S^{va}_{sub} are the largest difference, these values much are in the border to distinguish between normal and abnormal data. Let s_t and s_{t+1} be the two consecutive values in S^{va}_{sub} and that $s_{t+1} - s_t$ is the greatest value compared to all other pairs, then the threshold is dertermined as the mean of s_t and s_{t+1}. More details are presented in Algorithm 1.

[1] Here $len()$ function returns the number of samples in S^{va}. Two values 0.90 and 0.98 are used since we assume that from 90% to 98% of samples are normal.

Algorithm 1 Threshold estimation for OCCs

Given classifier f_j, validation set va.
$S^{va} \leftarrow sort(f_i(va))$
$S^{va}_{sub} \leftarrow subset(S^{va})$
$threshold \leftarrow 0$
$vol \leftarrow 0$
for $k \leftarrow 0$ to $|S^{va}_{sub}| - 1$ **do**
    $dif \leftarrow S^{va}_{sub}[k+1] - S^{va}_{sub}[k]$
    if $dif > vol$ **then**
        $vol \leftarrow dif$
        $threshold \leftarrow (S^{va}_{sub}[k+1] + S^{va}_{sub}[k])/2$
return $threshold$

4.2. Metric to measure the generalization of OCCs

In fusion learning, each individual classifier should contribute to the final model corresponding to the weight w.r.t their generalization ability [9]. In this section, we propose a novel approach for measuring the generalization ability of OCCs without using anomaly data. Let $train$ and va be the training set and the validation set, respectively. A given classifier f_j has a good generalization ability if its performance on the $train$ is approximate to that on va. Our idea is to define a generalization error (gen_error_j) for f_j as the average difference between the classification accuracy of f_j on the $train$ and va over several thresholds t^i. First, several thresholds t^i are chosen. With these thresholds, the true negative rates of f on $train$ and val are TNR^i_{train} and TNR^i_{va}, respectively. In this work, we choose nine thresholds in the range of $[90.0\%, 98.0\%]$ with an interval of 1.0% resulting in nine values of TNR_{train} and TNR_{va} on the $train$ and va. Finally, the mean absolute errors between the accuracy of f_j on the $train$ and va at these thresholds are calculated as Eq. (11)

$$gen\_error_j = \frac{1}{k}\sum_{i=1}^{k}\left|TNR^i_{train} - TNR^i_{va}\right|, \tag{11}$$

where k is the number of classification thresholds and t^i is the i-th threshold. Table 1 shows gen_error_j at the nine thresholds of four OCCs on the NSL-KDD dataset.

4.3. Calculating the weight of OCCs

In this section, the weight of each OCC in the fusion model is calculated based on its generalization error. A classifier with a smaller gen_error should have a larger contribution to the fusion-based model. The weight of a classifier f_j can be computed and normalized in the range of $[0, 1]$ w.r.t the weight of other classifiers in the model as in Eq. (12)

$$w_j = \frac{\min\left[gen\_error_1, gen\_error_2.., gen\_error_m\right])}{gen\_error_j}, \tag{12}$$

where m is the number of one-class classifiers in the fusion model.

4.4. Estimating mass function for hypothesis

In fusion learning, a Basic Probability Assignment (BPA) function is used to assign mass values (the belief committed) to each hypothesis in the hypothesis space. In anomaly detection, the hypothesis space, $P(\Theta) = \{A, N, NA, \emptyset\}$, where A and N are abnormal and normal hypotheses, $N \cap A = \emptyset$ and NA is an uncertain hypothesis. In this paper, we develop a new BPA function called the One-class Basic Probability Assignment function (OBPA), to estimate mass values for each data sample. For a given data point, OBPA takes its anomaly score produced by an OCC f_j as the input. The corresponding output of OBPA is the mass values assigned for each hypothesis in $P(\Theta)$ as $m(A)$, $m(N)$, $m(\emptyset)$ [2], and $m(NA)$ for the data point.

To estimate these values, we first determine the anomaly scores S^{va} of the samples in va using f_j. Next, we introduce two deviations, d_N and d_A, for the N and A areas, respectively. d_N is defined as the distance from the smallest anomaly score in S^{va} to the threshold t (t is estimated in Subsection 4.1), while d_A is the distance from threshold t to the maximum anomaly score in S^{va}. By introducing a bandwidth, bw [3], we can determine the uncertain area on A as $\sigma_A = d_A \times bw$, and the uncertain area on N as $\sigma_N = d_N \times bw$. Thus, the uncertain area, NA, can be defined as $[t - \sigma_N, t + \sigma_A]$. Fig. 1 represents S^{val} the threshold t, the deviations d_N and d_A, and the areas σ_N and σ_A.

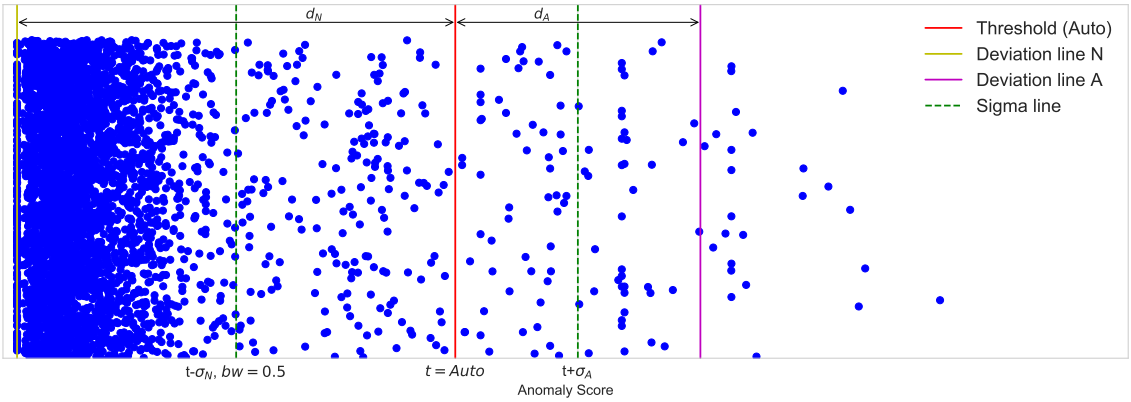


Figure 1: Choosing threshold (t) and deviation (σ) from the score of training data

Algorithm 2 describes the OBPA function to calculate $m(A), m(N), m(NA)$ for a given data sample x_i. When the anomaly score s_i is less than the threshold t, $m(A)$ is assigned to a high value and $m(N)$ is assigned to a very small value. Conversely, when s_i is greater than t, $m(A)$ is very small and $m(N)$ is high. In Algorithm 2, the input initial mass b_0 is set to 0.5 and a very small value 10^{-5} is set for u_0. Moreover, it is important to guarantee that $m(A) + m(NA) + m(N) = 1$ and $m(\emptyset) = 0$ as in Eq. (2).

[2]Note that $m(\emptyset) = 0$

[3]In our paper, $bw = 0.5$ is calibrated from the experiments.

Algorithm 2 The OBPA function

procedure (anomaly score s_i, threshold t, basic belief b_0, unbelief u_0, deviation σ_N, σ_A, and bandwidth bw)
  if $s_i \leqslant t$ **then**
    $b_1 \leftarrow (t - s_i)/\sigma_N$;
    $m(N) \leftarrow b_1 * b_0$;
    $m(A) \leftarrow u_0$;
    $m(NA) \leftarrow (1 - mN - mA)$;
  else
    $b_1 \leftarrow (s_i - t)/\sigma_A$;
    $m(A) \leftarrow b_1 * b_0$;
    $m(N) \leftarrow u_0$;
    $m(NA) \leftarrow (1 - m(A) - m(N))$;
  return $m(A), m(N), m(NA)$

4.5. One-class fusion-based anomaly detection model

This section presents how to fuse single OCCs to form the final model, i.e., OFuseAD. Fig. 2 shows the fusing stage of our proposed model. First, four OCCs (i.e. LOF, $DSAE_{Z^1}$, $DSAE_{Z^2}$, and S_{KDE}) trained on the normal training data are used to computed anomaly scores (i.e. S_{LOF}, $S_{DSAE_{Z^1}}$, $S_{DSAE_{Z^2}}$, and S_{KDE}) for the samples in the validation set va. After that, the threshold, the generalization error, and the weight for each OCC are estimated using the methods presented in Subsections 4.1, 4.2, and 4.3, respectively. Next, the OBPA function is applied on each of these OCCs to estimate the mass value for each data sample. The final step is to combine the mass values from the four OCCs to create the final mass value for OFuseAD. The final mass value is calculated using the Dempster-Shafer's rule of Combination (DRC).

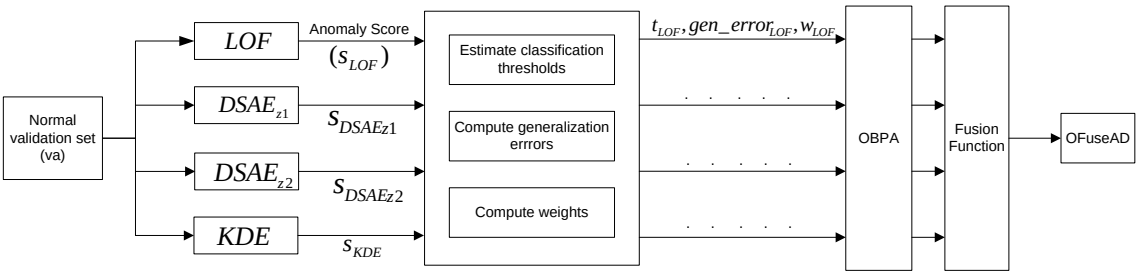


Figure 2: The fusion phase of our proposed model, OFuseAD

$$m(H) = \frac{\sum_{(B \cap C = H)} [w_i m_i(B) * w_j m_j(C)]}{1 - \sum_{(B \cap C = \varnothing)} [w_i m_i(B) * w_j m_j(C)]} \tag{13}$$

where m_i and m_j are the masses produced from the i-th classifier and the j-th OCC. w_i and w_j are the weights for i and j-th OCC, respectively. B and C are the hypotheses in Θ supported by the source E_i and E_j, respectively. Using Eq. (13), we calculate the the mass

value $m(N)$, $m(A)$ and $m(NA)$ for each query sample. If the mass $m(N)$ is higher than $m(A)$, the state of the system will be is normal, otherwise it is abnormal.

5. EVALUATION AND DISCUSSION

5.1. Data sets

In our experiments, ten well-known datasets in the network security domain are employed for evaluating the model.

- The NSL-KDD dataset is a filtered version of the KDD Cup'99 dataset [32]. It is split into two parts: The training set of 67343 examples and the testing set of 9711 normal examples and 12833 anomalies. Each connection consists of 41 features with a label (normal or an attack name).

- The UNSW-NB15 dataset [33] was aimed to solve some drawbacks in the previous public datasets. UNSW-NB15 consists of a training set of 56000 records and a testing set of 37000 normal instances and 45332 anomalies. Each samples is constructed from 47 features with a normal label or an attack label.

- CTU13 [34] is a botnet dataset that consists of thirteen botnet scenarios. In this paper, we only use four scenarios, namely CTU13_08, CTU13_09, CTU13_10, and CTU13_13. They are split each of them into 40% for training (only normal) and 60% for evaluating.

- BoT-IoT was created from a realistic network environment in the Cyber Range Lab of the center of UNSW Canberra Cyber [35]. The dataset contains DDoS, DoS, OS, Service Scan, Key-logging and Data ex-filtration attacks. We use the extracted version of the dataset with about 3 million records. We sample 20% of normal data for training and 80% of normal and anomalies for testing.

- The Spambase and the InternetADs datasets are from the UCI machine learning repository [36]. The Spambase is a set of spam e-mails, constructed from 57 features. The InternetADs dataset represents a collection of possible advertisements on Internet pages. It is constructed from 1558 features with 1582 records.

- WUSTL-IIOT-2018 is a dataset used for SCADA cybersecurity research [37]. The dataset was built using the SCADA system testbed described in [37] to emulate real-world industrial systems. It is constructed from 6 features with 505921 normal examples and 13750 anomalous records. The 10% of normal data is selected for training and the rest is for testing.

The categorical features in these datasets are processed by using one-hot-encoding resulting in higher dimension versions. All features are then normalized into $[-1, 1]$.

5.2. Parameter settings

To create the fusion model, we used three individual OCCs including a deep learning model (Double-Shrink AutoEncoder-DSAE [23]), a distance-based method (Local Outliner Factor-LOF [22]), and a density-based method (Kernel Density Estimation-KDE [20]).

Table 1: Generalization errors and weight of four OCCs on NSL-KDD

AD Methods	The generalization errors at classification thresholds									Mean errors	Weighted scores
	90%	91%	92%	93%	94%	95%	96%	97%	98%		
$DSAE_Z1$	0.0001	0.0009	0.0018	0.0015	0.0002	0.0002	0.0005	0.0025	0.0034	0.0012	1.0000
$DSAE_Z2$	0.0007	0.0026	0.0005	0.0005	0.0012	0.0012	0.0039	0.0034	0.0036	0.0020	0.5952
LOF	0.0076	0.0057	0.0040	0.0029	0.0056	0.0051	0.0051	0.0056	0.0025	0.0046	0.2577
KDE	0.0009	0.0009	0.0005	0.0006	0.0001	0.0008	0.0023	0.0034	0.0053	0.0016	0.7407

For DSAE, we use two versions with latent representations Z^1 and Z^2 and they are called $DSAE_{Z^1}$ and $DSAE_{Z^2}$, respectively. The hyper-parameters are set for the OCCs as follows

- The Gaussian kernel is used for KDE, and its scaling parameter γ is set by a default value, $\gamma = 0.1$ [38]. The number of nearest neighbors in LOF, k, is chosen as 1% of the training size.

- The architecture and hyper-parameters of DSAE are configured as in the previous work. Specifically, the weights of DSAE are initialized following [39] and the activation function used in DSAE is *tanh*. We train the model over 1000 epochs using Adadelta.

To evaluate the performance of OFuseAD, we used three popular metrics including F1-score (F1), Accuracy (ACC), and the Area Under the ROC Curves (AUC). All experiments are implemented in Python 3.7 and operated on a machine with the configuration as follows: CPU Intel Core i5, 8GB RAM, and RAM frequency of 1867MHz. OFuseAD and other OCCs are implemented using TensorFlow Library [40]. LOF and KDE provided by scikit-learn are employed [41].

6. RESULTS AND DISCUSSION

We divided our experiments into two sets. The first set is to estimate the threshold of each OCC and its weight in the final fusion model. The generalization error and the weight of each OCC on ten datasets are presented in Table 1.

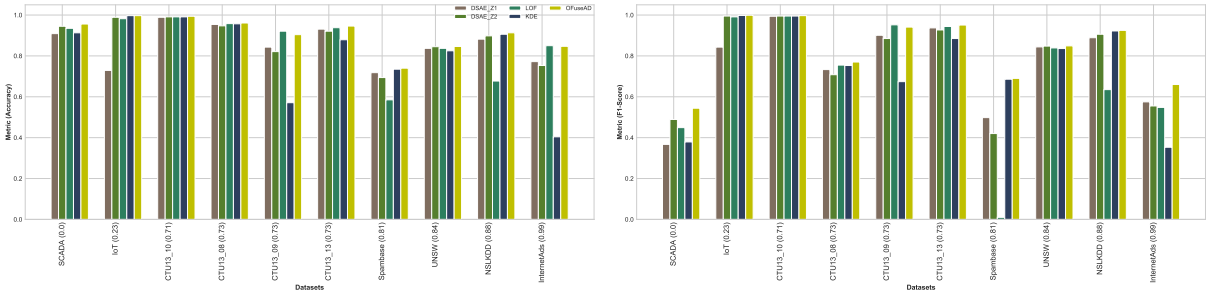
The second set of experiments is to evaluate the effectiveness of OFuseAD and to compare it with the single models. The accuracy, F1-score, and AUC of OFuseAD and the four single OCCs are shown in Tables 2, 3 and 4 and Figs. 3(a), 3(b), and 4. It can be seen from these tables and figures that OFusedAD achieves better performance than the four single OCCs on almost every dataset (except CTU13_09) when using F1-Score and ACC metrics. For example, the accuracy of OFuseAD on InternetAds is 0.847 compared to 0.772, 0.753, 0.850, and 0.404 of $DSAE_{Z^1}$ and $DSAE_{Z^2}$, LOF, and DKE, respectively. Similarly, the F1-score of OFuseAD on SCADA is improved from 0.367, 0.489, 0.449, and 0.379 of $DSAE_{Z^1}$ and $DSAE_{Z^2}$, LOF, and DKE, respectively, to 0.544. When using the AUC metric, the result of OFuseAD is not as convincing as using accuracy and F1-score. This is because OFuseAD can find a specific threshold to well separate between normal and abnormal data on each dataset. Thus, the value of ACC and F1-score of OFuseAd is better than those of the others. However, when averaging the result on multiple thresholds in AUC, the performance of OFuseAD is not much different from the other single OCCs.

Table 2: Accuracy of individual one-class classifiers and OFuseAD on ten datasets

Methods	Datasets									
	SCADA (0.0)	IoT (0.38)	CTU13_10 (0.71)	CTU13_08 (0.73)	CTU13_09 (0.73)	CTU13_13 (0.73)	Spambase (0.81)	UNSW (0.84)	NSLKDD (0.88)	InternetAds (0.99)
$DSAE_Z1$	0.909	0.729	0.989	0.954	0.843	0.931	0.718	0.837	0.882	0.772
$DSAE_Z2$	0.945	0.989	0.991	0.947	0.821	0.921	0.694	0.845	0.898	0.753
LOF	0.935	0.982	0.991	0.958	0.921	0.938	0.585	0.837	0.677	0.850
KDE	0.913	0.997	0.991	0.957	0.571	0.879	0.735	0.825	0.906	0.404
OFuseAD bw=0.5	**0.956**	**0.997**	**0.994**	**0.961**	0.904	**0.946**	**0.740**	**0.846**	**0.913**	0.847

Table 3: F1 Score from individual one-class classifiers and OFuseAD on ten datasets

AD Methods	Datasets									
	SCADA (0.0)	IoT (0.38)	CTU13_10 (0.71)	CTU13_08 (0.73)	CTU13_09 (0.73)	CTU13_13 (0.73)	Spambase (0.81)	UNSW (0.84)	NSLKDD (0.88)	InternetAds (0.99)
$DSAE_Z1$	0.367	0.843	0.994	0.733	0.901	0.937	0.498	0.844	0.889	0.575
$DSAE_Z2$	0.489	0.995	0.995	0.708	0.885	0.927	0.420	0.848	0.906	0.555
LOF	0.449	0.991	0.995	0.755	0.952	0.944	0.010	0.839	0.635	0.548
KDE	0.379	0.998	0.995	0.753	0.674	0.885	0.686	0.836	0.922	0.353
OFuseAD bw=0.5	**0.544**	**0.999**	**0.997**	**0.770**	0.941	**0.951**	**0.690**	**0.849**	**0.925**	**0.660**



(a) (b)

Figure 3: The Visualization of (a) classification accuracy and (b) F1 score from OFuseAD again four classifiers.

Figure 4 visualizes the ROC curve of each single OCC and the OFuseAD model. In this paper, we just show the AUC on two datasets, CTU13-08 and CTU13-13, for analysis purposes. It can see that although the AUC of OFuseAD is not always higher than those of the individual models, the ROC curves of OFuseAD are still approximately 0.1% higher than the three single models.

Table 4: AUCs from individual one-class classifiers and OFuseAD on ten datasets

AD Methods	Datasets									
	SCADA (0.0)	IoT (0.38)	CTU13_10 (0.71)	CTU13_08 (0.73)	CTU13_09 (0.73)	CTU13_13 (0.73)	Spambase (0.81)	UNSW (0.84)	NSLKDD (0.88)	InternetAds (0.99)
$DSAE_Z1$	0.991	0.920	0.994	0.985	0.942	0.966	0.840	0.882	0.966	0.958
$DSAE_Z2$	0.991	0.956	0.992	0.986	0.931	0.972	0.827	0.903	0.963	0.959
LOF	0.993	0.967	0.999	0.982	0.973	0.988	0.743	0.893	0.850	0.831
KDE	0.991	0.999	0.999	0.985	0.808	0.944	0.818	0.888	0.938	0.927
OFuseAD bw=0.5	0.991	0.999	0.996	**0.991**	0.949	0.980	0.831	**0.906**	0.965	0.946

One of the reasons for the effectiveness of OFuseAD is due to its ability to automatically find the good threshold for each OCC. For instance, an OCC can achieve high performance at a decision threshold t^1, but another OCC can obtain its peak of performance at the threshold t^2. If we use only one decision threshold, say t^1, for the two OCCs, only the first OCC can achieve high performance while the second OCC does not. Fortunately, OFuseAD can find a good threshold for each single OCC and aggregate them into one model to achieve better F1-Score and Accuracy in comparison to all single OCCs.

Overall, our proposed model, OFuseAD, inherits the advantages of the single one-class classifiers to perform robustly on almost all datasets. The method to automatically estimate the classification threshold for OCCs allows OFuseAD to fuse single OCC at decision thresholds in which these classifiers produce the highest performance. Consequently, the resulting fusion-based model, OFuseAD, can often produce a higher F1-score and ACC than the single OCC. Moreover, OFuseAD can eliminate the difficulty in calculating the weight of each OCC in the fusion model. Thus, OFuseAD is more suitable for deploying to a real-world network than using single models where the demand for the accuracy of detection is greater than the speed of detection.

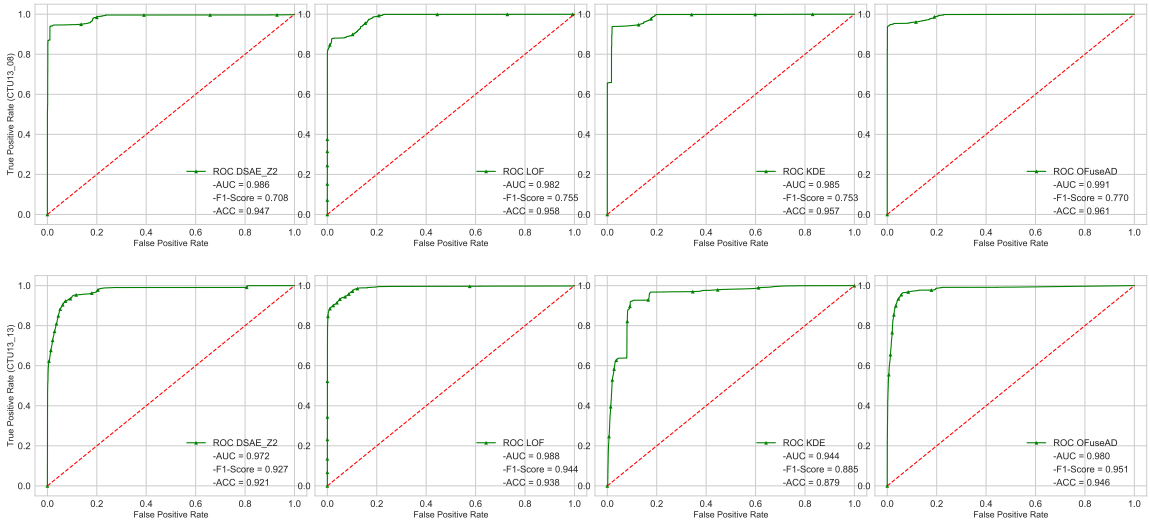


Figure 4: The ROC curves of three classifiers in comparison to that of OFuseAD

7. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a one-class fusion-based anomaly detection model, called OFuseAD. As far as we know, our research is the first attempt to develop a one-class fusion-based anomaly detection model based on the DS theory. We have proposed solutions to overcome some difficulties in fusing one-class classifiers where anomaly data is not available for estimating the thresholds and determining the weight of OCCs. We then apply the new version of the DS theory to construct a one-class fusion-based anomaly detection model for network anomaly detection.

OFuseAD is evaluated on ten datasets in the field of network security, and its performance

is better than the four single models. Moreover, our solution can work without manually setting the threshold, which security experts often do. Thus, this solution meets the requirement of deploying to a real-world network. In the future, we intend to carry out extensive experiments to investigate the performance of the OFuseAD on different domains and compare it to other well-known research methods. We also define a new DS theory's FoD for the system state that has more than two states normal and anomaly, such as normal and some specific attack types.

ACKNOWLEDGEMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2019.05.

REFERENCES

[1] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J Netw Comput Appl*, vol. 60, pp. 19–31, 2016.

[2] V. Bonandrini, J.-F. Bercher, and N. Zangar, "Machine learning methods for anomaly detection in IoT networks, with illustrations," in *International Conference on Machine Learning for Networking*. Springer, 2019, pp. 287–295.

[3] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[4] B. Sharma, L. Sharma, and C. Lal, "Anomaly detection techniques using deep learning in IoT: A survey," in *ICCIKE*. IEEE, 2019, pp. 146–149.

[5] V. L. Cao, M. Nicolau, and J. McDermott, "Learning neural representations for network anomaly detection," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 3074–3087, 2019.

[6] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[7] L. Vu, V. L. Cao, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Learning latent distribution for distinguishing network traffic in intrusion detection system," in *2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[8] P. Gogoi, D. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *The Computer Journal*, vol. 54, no. 4, pp. 570–588, 2011.

[9] Y. Liu, X. Wang, and K. Liu, "Network anomaly detection system with optimized DS evidence theory," *The Scientific World Journal*, vol. 2014, 2014.

[10] J. Kaliappan, R. Thiagarajan, and K. Sundararajan, "Fusion of heterogeneous intrusion detection systems for network attack detection," *The Scientific World Journal*, vol. 2015, 2015.

[11] T. C. Bui, M. Hoang, Q. U. Nguyen, and C. L. Van, "Data fusion-based network anomaly detection towards evidence theory," in *Proc. NICS*. IEEE, 2019, pp. 33–38.

[12] G. Li, Z. Yan, Y. Fu, and H. Chen, "Data fusion for network intrusion detection: A review," *Security and Communication Networks*, vol. 2018, pp. 1–16, 05 2018.

[13] J. Tian, W. Zhao, and R. Du, "DS evidence theory and its data fusion application in intrusion detection," in *International Conference on Computational and Information Science*. Springer, 2005, pp. 244–251.

[14] K. S. M. Raja and K. J. Kumar, "Diversified intrusion detection using various detection methodologies with sensor fusion," in *Proc. ICCPEIC*. IEEE, 2014, pp. 442–448.

[15] V. Shah, A. K. Aggarwal, and N. Chaubey, "Performance improvement of intrusion detection with fusion of multiple sensors," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 33–39, 2017.

[16] Q. Chen, A. Whitbrook, U. Aickelin, and C. Roadknight, "Data classification using the dempster–shafer method," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 26, no. 4, pp. 493–517, 2014.

[17] A. Mattar and M. Z. Reformat, "Detecting anomalous network traffic using evidence theory," in *Advances in Fuzzy Logic and Technology 2017*. Springer, 2017, pp. 493–504.

[18] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976, vol. 42.

[19] J. Dezert, A. Tchamova, and F. Dambreville, "On the mathematical theory of evidence and dempster's rule of combination," 2011.

[20] M. P. Wand and M. C. Jones, *Kernel Smoothing*. Chapman and Hall/CRC, 1994.

[21] V. L. Cao, M. Nicolau, and J. McDermott, "A hybrid autoencoder and density estimation model for anomaly detection," in *Proc. PPSN*. Springer, 2016, pp. 717–726.

[22] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 2000, pp. 93–104.

[23] C. T. Bui, L. C. Van, M. Hoang, and Q. U. Nguyen, "A double-shrink autoencoder for network anomaly detection," *Journal of Computer Science and Cybernetics*, vol. 36, no. 2, pp. 159–172, 2020.

[24] T. Bass, "Intrusion detection systems and multisensor data fusion: Creating cyberspace situational awareness," *Communications of the ACM*, vol. 43, no. 4, pp. 99–105, 2000.

[25] G. Giacinto, F. Roli, and L. Didaci, "Fusion of multiple classifiers for intrusion detection in computer networks," *Pattern Recognit. Lett.*, vol. 24, no. 12, pp. 1795–1803, 2003.

[26] Y. Wang, H. Yang, X. Wang, and R. Zhang, "Distributed intrusion detection system based on data fusion method," in *Proc. WCICA*, vol. 5. IEEE, 2004, pp. 4331–4334.

[27] C. Siaterlis and B. Maglaris, "Towards multisensor data fusion for DoS detection," in *Proceedings of the 2004 ACM Symposium on Applied Computing*. ACM, 2004, pp. 439–446.

[28] W. Hu, J. Li, and Q. Gao, "Intrusion detection engine based on dempster-shafer's theory of evidence," in *2006 International Conference on Communications, Circuits and Systems*, vol. 3. IEEE, 2006, pp. 1627–1631.

[29] C. Thomas and N. Balakrishnan, "Improvement in intrusion detection with advances in sensor fusion," *IEEE Trans. Inf. Forensics Secur*, vol. 4, no. 3, pp. 542–551, 2009.

[30] X. Zhao, H. Jiang, and L. Jiao, "A data-fusion-based method for intrusion detection system in networks," *IJIEEB*, vol. 1, no. 1, p. 32, 2009.

[31] C. Lu, Y. Li, M. Ma, and N. Li, "A hybrid NIDS model using artificial neural network and DS evidence," *IJDCF*, vol. 8, no. 1, pp. 37–50, 2016.

[32] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE, 2009, pp. 1–6.

[33] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *Proc. MilCIS*. IEEE, 2015, pp. 1–6.

[34] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers & Security*, vol. 45, pp. 100–123, 2014.

[35] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: BoT-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[36] A. Asuncion and D. Newman, "UCI machine learning repository," 2007.

[37] M. A. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, "Scada system testbed for cybersecurity research using machine learning approach," *Future Internet*, vol. 10, no. 8, p. 76, 2018.

[38] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*, 2018, pp. 4393–4402.

[39] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[40] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. OSDI 16*, 2016, pp. 265–283.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *JMLR*, vol. 12, no. Oct, pp. 2825–2830, 2011.

Received on October 31, 2021
Accepted on January 09, 2023