Online Intrusion Detection for IoT Systems with Full Bayesian Possibilistic Clustering and Ensembled Fuzzy Classifiers

Fang-Qi Li, Rui-Jie Zhao, †Shi-Lin Wang, *Senior Member, IEEE,* Li-Bo Chen,
Alan Wee-Chung Liew, *Senior Member, IEEE,* Weiping Ding, *Senior Member, IEEE*

***Abstract*—The pervasive deployment of the internet of things (IoT) has significantly facilitated manufacturing and living. The diversity and continual updates of IoT systems make their security a crucial challenge, among which the detection of malicious network traffic turns out to be the most common yet destructive threat. Despite the efforts on feature engineering and classification backend designing, established intrusion detection systems sometimes lack robustness and are inflexible against the shift of the traffic distribution. To deal with these disadvantages, we design a fuzzy system for the online defense of IoT. Our framework incorporates a full Bayesian possibilistic clustering module for feature processing and an ensemble module motivated by reinforcement learning and adaptive boosting that dynamically fits the streaming data. The proposed clustering module overcomes the issue of determining the number of clusters and can dynamically identify new patterns. The classifier backend combines a collection of fuzzy decision trees that provide readable decision boundaries. The ensembled classifiers can accommodate the drift of data distribution to optimize the long-time performance. Our proposal is tested on settings including one dataset collected from real IoT systems and is compared to numerous competitors. Experimental results verified the advantage of our system regarding accuracy and stability.**

***Index Terms*—IoT security, fuzzy clustering, ensemble learning.**

I. Introduction

THE development of internet of things (IoT) techniques including mobile device and edge computing algorithms, is boosting industrial IoT applications. By connecting heterogeneous sensors through decentralized networks and analyzing the collected data by cloud computing servers, IoT is revolutionizing modern industry and reshaping human life. By 2025, there are expected to be more than 64B IoT devices worldwide and the related industry is expected to generate \$4T to \$11T in economic value[1]. As a result, much effort has been devoted to fostering reliability and security in industrial IoT [1], [2].

Due to the limitation of computing source and the demand for an immediate response, the security in IoT is usually reduced to the privacy-preserving of data against semi-honest yet curious nodes within the IoT [3], the robustness against the compromise of edge devices [4], etc.

Among all the threats and challenges, the malicious intrusion to IoT systems remains the most common sabotage and the focus of current studies [5]. Many IoT terminals are fragile to cyber or physical attacks. It is estimated that fewer than 42% organizations can identify insecurity in their IoT devices, which are under attack every 5 minutes [2]. Consequently, these compromised terminals, with access to the cloud computing centers, can produce malicious traffic that finally ruins the entire system. For example, an adversary spoiling multiple edge device can conduct the distributed denial of service (DDoS) attack while an adversary impersonating a legitimate device on the network can obtain free access to the protected appliance [6]. Since increasing the security level of all edge devices is expensive, the responsibility of detecting intrusions and rebooting the compromised terminals is left for the cloud security center that eavesdropping on the IoT traffic.

For devices such as a modem or wireless router, the traffic contains evidence for intrusion detection which can be revealed from side channels [7]. Chen *et al.* demonstrated how to detect malicious traffic by identifying shared keywords in messages [8]. To analyze the general encrypted traffic in IoT, a collection of features is distilled from transmitted packages. Some representative toolkits for feature extraction are Argus[3], Bro-IDS[4], and Tranalyzer2[5]. Having obtained the statistics, many preprocessing models can be leveraged to select versatile components, reduce dimensionality, or produce clusters.

Identifying malicious traffic is essentially a classification task. Therefore, many classification models, including expert experience-based rules [9] and machine learning-based systems [5], [10] have been proposed as the intrusion detection

F. Li, R. Zhao, S. Wang, and L. Chen are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. (e-mail: solour_lfq@sjtu.edu.cn; ruijiezhao@sjtu.edu.cn; wsl@sjtu.edu.cn; bob777@sjtu.edu.cn)

A. W. Liew is with the School of Information and Communication Technology, Griffth University, Gold Coast Campus, QLD 4222, Australia. (e-mail: a.liew@griffth.edu.au).

W. Ding is with the School of Information Science and Technology, Nantong University, Nantong 226019, China. (e-mail: ding.wp@ntu.edu.cn)

†Shi-Lin Wang is the corresponding author.

The work described in this paper was supported in part by the National Natural Science Foundation of China (61771310, 61300167, 61976120), the Natural Science Foundation of Jiangsu Province under Grant BK20191445, the Natural Science Key Foundation of Jiangsu Education Department under Grant 21KJA510004, the Qing Lan Project of Jiangsu Province, and the Science and Technology Commission of Shanghai Municipality Research Program under Grant 20511102002. Our gratitude goes to the anonymous reviewers for their efforts.

Manuscript received October 31, 2021.

[1] https://techjury.net/blog/internet-of-things-statistics/

[2] https://www.thesslstore.com/blog/20-surprising-iot-statistics-you-dont-already-know/

[3] http://qosient.com/argus/index.shtml

[4] https://www.bro.org/index.html

[5] https://tranalyzer.com/downloads.html

backend. Intrusion detection systems (IDS) can be divided into misuse-based ones and anomaly-based ones [11]. An anomaly-based IDS profiles benign traffic and computes the distance between a suspicious message and benign history to form alerts [12]. A misuse-based IDS extracts characteristic features or signatures from traffic data. By comparing the signature of the message, the IDS evaluates the message's threat.

In practice, IoT is confronted with numerous zero-day attacks [13], whose distribution is unpredictable. Therefore, misuse-based IDSs that explicitly differentiate normal traffic from intrusion is the more reliable option. Yet the diversified patterns within the traffic and the complexity due to device heterogeneity increase the difficulty in designing IDSs for IoT. After incorporating new appliances into the IoT, the distribution of the traffic might change. This change is known as the *concept drift*, a challenge for online and even life-long learning [14]. Since most IDSs cannot efficiently fit streaming data on the fly, such an update would result in either a high false-positive rate or a slow reaction speed. Meanwhile, many machine learning-based models, especially deep learning-based ones, sacrifice interpretability, which is a prerequisite for reliable and robust security [15]. Instead, fuzzy systems, which is usually a combination of fuzzy clustering [16], fuzzy logic [17], and fuzzy inference modules [18], turn out to be a promising candidate in yielding robust and explainable decision boundaries with a manageable error rate.

In this paper, we leverage fuzzy clustering and fuzzy logic to construct a misuse-based IDS for IoT. To cope with outliers and the concept drift, we propose a full Bayesian variant of the possibilistic C-means clustering [19], one specific type of fuzzy clustering. To probe new patterns and determine the number of clusters, an evidence framework is designed for the proposed clustering algorithm. To generate interpretable classifiers, we adopt a fuzzy version of random forest and design a weighting scheme to combine adaptive boost with online learning to accommodate the concept drift. The contributions of our paper are threefold:

- We design the full Bayesian possibilistic C-means clustering (FBPCM) with configurable prior for the membership. It is robust against outliers and noise. An evidence framework is designed to determine the number of clusters and mines new centroids dynamically.
- We propose an online ensemble to combine base fuzzy rule classifiers. By assigning different weights for samples adaptively, the ensembled classifier backend preserves good performance under the change of the traffic's underlying distribution.
- We collected a new dataset from a live IoT and compared our scheme with other IDSs on it and other public datasets. Experimental results justified the privilege of our proposal regarding accuracy and stability.

The rest of the paper is organized as follows: Section II summarizes the related works. Our method is proposed in Section III. Section IV presents the experiments and discussions. Finally, Section V concludes the paper.

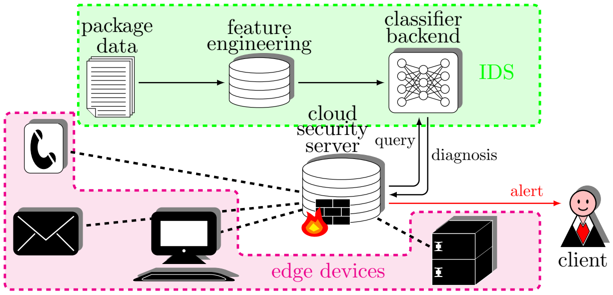


Fig. 1. IDS in IoT system.

II. Related Works

A. Intrusion detection in IoT

Numerous IDSs have been designed for network traffic. They are usually composed of a feature engineering module and a classifier backend as shown in Fig. 1. Meidan *et al.* adopted an autoencoder to detect botnet attacks in IoT with their N-BaIoT [12]. In [6], Aminanto *et al.* leveraged an autoencoder to extract features for impersonation detection in Wi-Fi. Mirsky *et al.*'s work, Kitsune [20] applied ensembled autoencoder for online intrusion detection. Apart from autoencoders, recurrent neural networks were also adopted for intrusion detection [21], while Jiang *et al.* proposed a hybrid scheme with hierarchical network [22]. Moustafa *et al.* examined anomaly detection in large networks by modeling the statistics of traffic with a mixed beta distribution and updating the model with expectation-maximization [11]. Some recent works also combine IDS with IoT techniques such as blockchain to promote its reliability [23].

Collecting a network traffic dataset is an expensive task. Most network traffic datasets are collected from a local network as a sandbox, where attacks are conducted by simulation. Although most works on IDS for network traffic were evaluated on KDD'99 Dataset [24], it was reported that KDD'99 suffers from many problems such as missing attributes, the difference in distributions across the training and testing dataset, and obsolescence. To cope with these defects, Nour *et al.* collected UNSW-NB15 [25] as a substitute, they utilized the IXIA PerfectStorm tool [6] and eavesdropped on a simulation network for 31 hours. Then features ranging from flow, content, time, etc. were extracted using Argus and Bro-IDS (which was used to process KDD'99). Recently, new tools that parse network packages are providing more diversified and informative features [26]. A summary of features of interest and their properties is given in Table I. To evaluate the performance of an IDS fairly, it is necessary to combine the latest parsing tools with real network traffic.

Current misuse-based IDSs are known to have a high false-alarm rate due to poor adaptivity and class imbalance. Consequently, fuzzy systems have been adopted to implement robust and readable IDSs [27]. Yu *et al.* combined naive Bayes classifier with fuzzy logic for intrusion detection [28]. Fuzzy rules were widely used to reduce the false alarm in different network settings [29]. Abadeh *et al.* studied the efficacy of

[6] http://www.ixiacom.com/products/perfectstorm

TABLE I
SOME REPRESENTATIVE FEATURES OF NETWORK TRAFFIC.

Name	Description	Domain
`duration`	length of the connection	real
`service`	service protocol on the destination	categorical
`sttl`	source to destination time to live	integer
`numPktSnt`	number of transmitted packets	integer
`bytAsm`	byte stream asymmetry	real
`tcpEcI`	TCP estimated counter increment	real
`Bwd IAT Max`	maximum time between two packets sent in the backward direction	integer
`Subflow Bwd Bytes`	the average number of bytes in a sub flow in the backward direction	integer

different ways of generating fuzzy rules in IDS [30]. Fuzzy clustering, which has been extensively studied in domains as biomedical statistics [31], social studies [32], computer vision [33], and dynamics [31] is also incorporated into IDSs for pattern recognition [34], [35].

B. The concept drift

One implicit assumption behind machine learning-based IDSs is the identical distribution underlying the training dataset and the testing dataset. This assumption might fail due to the complexity of IoT traffic. For example, the change in the network topology or the setting of edge devices might modify the statistics of the traffic so models built on previous data are going to perform poorly. This phenomenon has been recognized in public datasets as KDD'99 [25].

Increasing the model's adaptivity and flexibility against the change of the data's distribution, known as the concept drift, is a crucial target for reinforcement learning, online learning, and life-long learning systems [36]. Ordinary solutions include detecting the concept drift through statistical testing and retraining the model [37], [38], updating the weights of ensembled classifiers w.r.t. the streaming data [39], etc.

These schemes have found successful outcomes in target identification [40], deploying of wireless sensor networks [41], etc. However, the concept drift in IDS, including identifying the change in traffic statistics and adaptively tuning the model, has not undergone sufficient studies [42], [43]. This is because established IDSs assume that traffic data are subject to a stationary distribution and no further tuning is necessary after the model's training. Consequently, they are fragile to attacks that are beyond the scope of the training data, whose frequency is high due to the complexity in IoT settings.

III. THE PROPOSED METHOD

A. The motivation

The multimodel distribution of IoT traffic statistics is better captured by clustering, which would yield human-readable patterns. Basic clustering algorithms such as C-means with one-hot membership [44] are known to be sensitive to outliers and noise. Fuzzy clustering algorithms abandon the one-hot constraint on the membership values and enable stronger representation capability. However, their robustness against noise remains ambiguous and unstable. Moreover, fuzzy semantics remains distinct from ordinary probability, making the incorporation of prior knowledge and the choice of optimal hyperparameter intractable. To evoke a probabilistic or even a Bayesian perspective of clustering, we proposed the Bayesian possibilistic C-means clustering (BPCM) [19], yet it is not a complete Bayesian framework since the prior distribution on the membership assignment is absent. To combine the robustness against the overwhelming noise in IoT traffic and the capability of the Bayesian method, we introduce the full Bayesian possibilistic C-means clustering (FBPCM) for pattern recognition in IoT traffic.

After identifying patterns from the features, we adopt an ensemble of fuzzy decision trees as the classifier backend. Fuzzy decision trees infer as fuzzy logic rules [45], [46] and are easy to train, robust, and straightforwardly interpretable. To adapt to concept drift in later samples, we modify the ordinary adaptive boosting [47] framework by assigning distinctive weights to different batches of samples. The recently arriving batches would have larger weights in updating the classifier accordingly.

B. Full Bayesian Possibilistic Clustering

1) Formulation: The variables used in FBPCM are listed in Table II. For hard clustering algorithms such as C-means, $\mathbf{u}_n$ is a one-hot vector. In fuzzy clustering it is subject to: $\sum_{k=1}^{K} u_{n,k}^{p} = 1$, while the constraint for possibilistic clustering is: $\sum_{n=1}^{N}(1-u_{n,k}^{p})^{\frac{1}{p}} = 1$. These constraints fuzzify the membership at the expense of complicating the parameter updating procedure. To fit the fuzzy clustering to the Bayesian framework, we continue to resort to a probabilistic setting. The likelihood of the observed data conditioned on the centroids is:

TABLE II
NOTATIONS OF INVOLVED VARIABLES.

Notation	Meaning
N	The number of samples in the dataset.
K	The number of clusters.
$\mathcal{X}$	The space of samples' feature.
$\mathbf{x}_n$	The n-th sample, $\mathbf{x}_n \in \mathcal{X}$.
$\mathbf{X}$	$\mathbf{X} = \{\mathbf{x}_k\}_{n=1}^{N}$ is the entire dataset.
$\mathbf{c}_k$	The k-th centroid, $\mathbf{c}_k \in \mathcal{X}$.
$\mathbf{C}$	$\mathbf{C} = \{\mathbf{c}_k\}_{k=1}^{K}$ is all centroids.
$u_{n,k}$	$u_{n,k} \geq 0$ is the membership of $\mathbf{x}_n$ to the k-th cluster.
$\mathbf{u}_n$	$\mathbf{u}_n = (u_{n,1}, u_{n,2}, \cdots, u_{n,K})^{\mathrm{T}}$.
d	$d : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty)$ is a distance metric.

$$
\begin{aligned}
\Pr(\mathbf{X}|\mathbf{C}) = \prod_{n=1}^{N} \Pr(\mathbf{x}_n|\mathbf{C}) &= \prod_{n=1}^{N} \int \Pr(\mathbf{x}_n, \mathbf{u}_n|\mathbf{C}) \mathrm{d}\mathbf{u}_n \\
&= \prod_{n=1}^{N} \int \Pr(\mathbf{x}_n|\mathbf{u}_n, \mathbf{C}) \cdot \Pr(\mathbf{u}_n) \mathrm{d}\mathbf{u}_n,
\end{aligned}
\tag{1}
$$

in which we assumed that the prior on $\mathbf{u}_n$ is independent of $\mathbf{C}$, i.e., the graphical model takes the form in Fig. 2.

To estimate parameters from a likelihood with the form of Eq. (1), it is necessary to adopt Expectation-Maximation (EM)

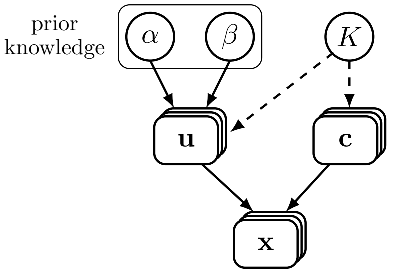


Fig. 2. The graphical model for FBPCM.

algorithm [48], where the expectation of $\mathbf{u}_n$ is firstly computed from the current centroids, then the variational lower bound of the likelihood is maximized by optimizing $\mathbf{C}$. To formulate the posterior distribution of $\mathbf{u}_n$, we begin with the following likelihood defined in BPCM [19]:

$$\Pr(\mathbf{x}_n|\mathbf{u}_n,\mathbf{C}) = \frac{1}{Z(\mathbf{u}_n,\mathbf{C})} \cdot \prod_{k=1}^{K} \exp\left\{-d(\mathbf{x}_n,\mathbf{c}_k)\right\}^{u_{n,k}}. \quad (2)$$

Notice that in Eq. (2), $u_{n,k}$ is essentially the inverse of the variance of a Gaussian distribution centered at $\mathbf{c}_k$ that generates $\mathbf{x}_n$. Therefore, the normalizer takes the form:

$$Z(\mathbf{u}_n,\mathbf{C}) = \prod_{k=1}^{K} \sqrt{\frac{2\pi}{u_{n,k}}}. \quad (3)$$

Since we do not exert any correlation between components of $\mathbf{u}_n$, Eq. (2) implies that the conjugate prior distribution for $u_{n,k}$ is a Gamma distribution:

$$\text{Gamma}(u_{n,k}|\beta,\alpha) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \cdot u_{n,k}^{\alpha-1} \cdot \exp\left\{-\beta \cdot u_{n,k}\right\}. \quad (4)$$

Combining Eq. (2), Eq. (3), and Eq. (4) suggests that the posterior distribution for $u_{n,k}$ is $\text{Gamma}(u_{n,k}|\beta',\alpha')$ with:

$$\alpha' = \alpha + \frac{1}{2},\ \beta' = \beta + d(\mathbf{x}_n,\mathbf{c}_k),$$

and its expectation becomes $\frac{\alpha'}{\beta'} = \frac{\alpha+\frac{1}{2}}{\beta+d(\mathbf{x}_n,\mathbf{c}_k)}$. Having finished the E-step, we substitute the results into Eq. (1) to compute the negative log auxiliary likelihood as:

$$\mathcal{L}(\mathbf{C}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}[u_{n,k}] \cdot d(\mathbf{x}_n,\mathbf{c}_k). \quad (5)$$

Optimizing Eq. (5) is straightforward since d is usually a convex distance metric. The EM update procedure for FBPCM is summarized in the following Algo. 1.

2) Determing the number of clusters: An intrinsic challenge for C-means clustering is determining the number of clusters, K. We suggest that selecting the optimal K in FBPCM is isomorphic to selecting the number of principal components in spectral analysis. Let the $K \times N$ matrix $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \cdots \mathbf{u}_N)$ denote the collection of all membership vectors. A representative cluster has a centroid, e.g. $\mathbf{c}_i$, that is close to a subset of $\mathbf{X}$, so the membership components in the corresponding i-th row in $\mathbf{U}$ are larger. For a mediocre centroid, e.g. $\mathbf{c}_j$, that lies in the neighbour of $\mathbf{c}_i$, the j-th row in $\mathbf{U}$ would be close to the i-th row and does not benefit the rank of $\mathbf{U}$. Therefore, the rank of $\mathbf{U}$, or its spectrum obtained from singular value decomposition, measures the appropriate K. Notice that hard C-means or ordinary fuzzy clustering does not preserve this property since they inherently differentiate different rows in $\mathbf{U}$, so $\mathbf{U}$'s spectrum can no longer embed unbiased information on K. This observation is illustrated in Fig. 3.

Algorithm 1 The EM procedure for FBPCM.

Input: The dataset $\mathbf{X}$, the number of clusters K, the distance metric d, prior parameters α and β, and the termination threshold ϵ.

Output: The centroids $\mathbf{C}$ and the membership values $\{\mathbf{u}_n\}_{n=1}^{N}$.

```
Randomly initialize $\mathbf{C}^{(0)}$, $t = 0$;
repeat
    for $n = 1$ to $N$ do
        for $k = 1$ to $K$ do
            $\alpha_{n,k} = \alpha + \frac{1}{2}$, $\beta_{n,k} = \beta + d(\mathbf{x}_n, \mathbf{c}_k^{(t)})$;
            $\mathbb{E}[u_{n,k}] = \frac{\alpha_{n,k}}{\beta_{n,k}}$;
        end for
    end for
    $++t$;
    for $k = 1$ to $K$ do
        $\mathbf{c}_k^{(t)} \leftarrow$ a minimizer of Eq. (5);
    end for
until $\|\mathbf{C}^{(t)} - \mathbf{C}^{(t-1)}\| \le \epsilon$
```

After adopting a full Bayesian perspective, the selection of K is reduced to the maximum a posteriori (MAP) estimation. The likelihood of K w.r.t. $\mathbf{X}$ is:

$$\Pr(\mathbf{X}|K) = \int \Pr(\mathbf{X},\mathbf{C}|K)\mathrm{d}\mathbf{C} \approx \Pr(\mathbf{X}|\mathbf{C}_K),$$

where $\mathbf{C}_K$ is K optimal centroids obtained by the EM procedure in Algo. 1 and the variational upper bound of this likelihood can be approximated by taking the expectation of the hidden variable $\mathbf{U}$. Therefore we have:

$$\log\Pr(K|\mathbf{X}) = \log\Pr(K) + \log\Pr(\mathbf{X}|\mathbf{C}_K) + \text{constant}. \quad (6)$$

The second term in Eq. (6) increases with K, while the first term embeds our prior knowledge on the number of clusters. Selecting the optimal K is tantamount to maximizing Eq. (6). The preference for fewer clusters can be incorporated into this formulation by introducing an appropriate prior, e.g., a geometric distribution, on K.

3) Evidence framework for concept drift: Since the data arriving at the IoT where the IDS has been deployed might vary with time, it is necessary to: (i) Update the configuration of the clusters w.r.t. the streaming data. (ii) Increase K to incorporate new patterns dynamically if necessary. The online version of EM given an arriving sample $\mathbf{x}_{(N+1)}$ firstly computes its membership to the k-th cluster, $u_{(N+1),k}$, as $\text{Gamma}(\beta + d(\mathbf{x}_{(N+1)},\mathbf{c}_k), \alpha + \frac{1}{2})$. Then each centroid $\mathbf{c}_k$ is updated by minimizing:

$$\mathcal{L}^{\text{new}}(\mathbf{c}_k) = \sum_{n=1}^{N} u_{n,k} \cdot d(\mathbf{x}_n,\mathbf{c}_k) + u_{(N+1),k} \cdot d(\mathbf{x}_{(N+1)},\mathbf{c}_k).$$

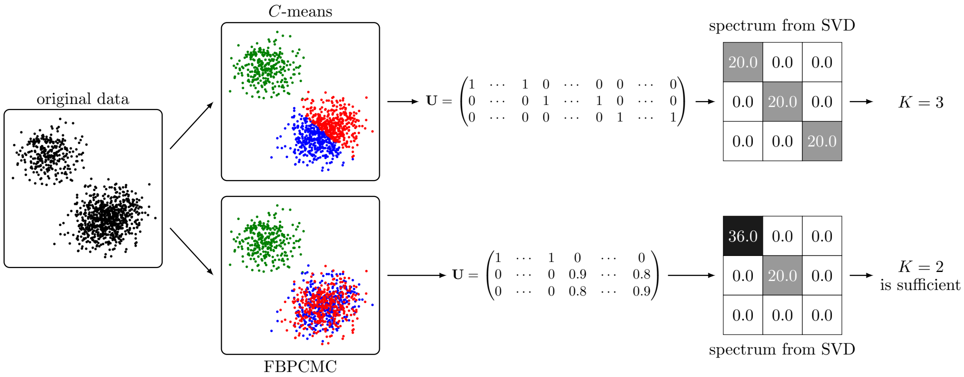

original data
C-means
FBPCMC
$\mathbf{U} = \begin{pmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & \cdots & 1 \end{pmatrix}$
spectrum from SVD
20.0 0.0 0.0
0.0 20.0 0.0
0.0 0.0 20.0
$K = 3$
$\mathbf{U} = \begin{pmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0.9 & \cdots & 0.8 \\ 0 & \cdots & 0 & 0.8 & \cdots & 0.9 \end{pmatrix}$
36.0 0.0 0.0
0.0 20.0 0.0
0.0 0.0 0.0
$K = 2$
is sufficient
spectrum from SVD

Fig. 3. Determining the number of clusters through SVD the membership matrix in ordinary C-means clustering and FBPCM.

In case where d is the l_2 loss, $\mathbf{c}_k$ is updated by:

$$\mathbf{c}_k^{\text{new}} \leftarrow \frac{\mathbf{c}_k \cdot \left(\sum_{n=1}^{N} u_{n,k}\right) + u_{(N+1),k} \cdot \mathbf{x}_{(N+1)}}{\sum_{n=1}^{N} u_{n,k} + u_{(N+1),k}},$$

and the membership values for previous data are updated as in the ordinary E-step. Having updated the centroids and memberships, the variational bound of the negative log likelihood conditioned on K can be estimated as:

$$\sum_{n=1}^{N+1} \sum_{k=1}^{K} \mathbb{E}[u_{n,k}] \cdot d(\mathbf{x}_n, \mathbf{c}_k^{\text{new}}) - \frac{1}{2} \ln \mathbb{E}[u_{n,k}]. \tag{7}$$

The assumption of identical distribution implies that Eq. (7) grows linearly. So if the speed of its increment exceeds a predefined threshold, it is likely that some unfamiliar pattern has appeared and a new $(K+1)$-th dimension should be introduced to the membership space, whose prior is given by:

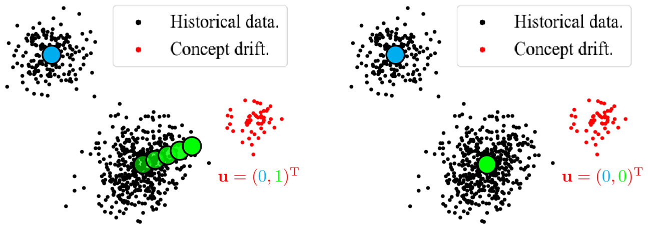
$$\alpha_{n,(K+1)} = \alpha,\ \beta_{n,(K+1)} = \frac{1}{\min_{k=1}^{K} \{\beta_{n,k}\}}, \tag{8}$$

so a sample who has already obtained a high membership to an established centroid is less likely to be contributed to the $(K+1)$-th cluster, while a sample who is distant from all existing clusters is likely to belong to the emerging cluster. After initialization, a few rounds of EM locate $\mathbf{c}_{(K+1)}$.

Remark: FBPCM does not push the membership assignment to unity by default, which is a desirable property for online clustering under the concept drift. It implies that the established centroids are more stable against the outliers, an example is given in Fig. 4. If the concept drift occurs then centroids in ordinary C-means clustering would drift accordingly since the membership of outliers to their closest cluster is high, even though they do not belong to this cluster. While centroids in FBPCM remain stable since the membership values of the samples from the new pattern are uniformly negligible.

C. Classifier

After analyzing the traffic features' statistics as clusters, a classifier backend produces predictions. The centroids of each dimension are marked as quantitative features and are fed into the classifier. Once the concept drift occurs, the classifier should efficiently incorporate the additional information by either profiling a new category of benign pattern or marking a malicious type of behavior. Our system includes a collection of fuzzy decision trees as basic classifiers and ensembles them under a sample reweighting scheme. Since the malicious traffic might swarm within a short time, e.g., in the DDoS attack, this temporal update is crucial for the stability of the IDS system. The overall framework is illustrated in Fig. 5.



(a) Ordinary C-means clustering.

(b) FBPCM.

Fig. 4. The drift of established centroids when concept drift takes place.

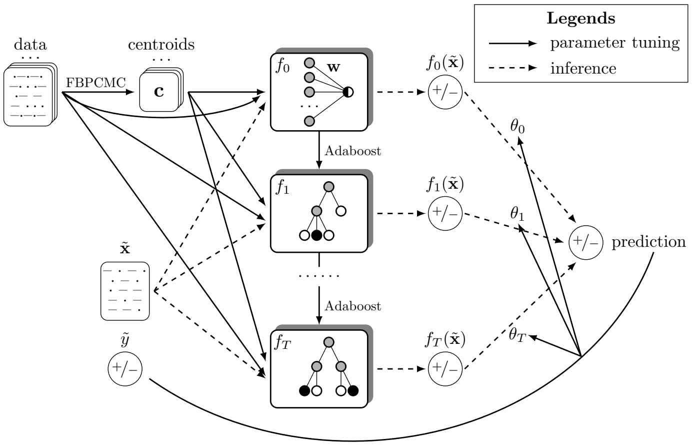


Fig. 5. The classifier backend.

1) Basic classifiers: The first basic classifier f_0 maps the deviation of a new observation, $\tilde{\mathbf{x}}$, from the established centroids, i.e., its membership vector, into an alert score. To ensure interpretability, f_0 is implemented as a linear discriminator with tunable parameter $\mathbf{w} = (w_0, w_1, \cdots, w_K)^{\mathrm{T}}$. The classification result is:

$$f_0(\tilde{\mathbf{x}}|\mathbf{w}) = \sigma(w_0 + \sum_{k=1}^{K} w_k^2 \cdot \tilde{u}_k),$$

where $\tilde{\mathbf{u}}$ is computed from an E-step within FBPCM, and $\sigma(\cdot)$ is the sigmoid function $\sigma(x) = \frac{1}{1+\exp\{-x\}}$. The positive output from f_0 denotes an alert for intrusion.

The membership assignments of benign traffic to established centroids are uniformly small since they occupy a dominating proportion of collected data. To reduce the false alarm rate, we exert an l_2 constraint on $\mathbf{w}$ during the training of f_0, whose loss function is formulated by:

$$\mathcal{L}_0(\mathbf{w}) = \sum_{n=1}^{N} \text{CE}(f_0(\mathbf{x}_n|\mathbf{w}), y_n) + \lambda \cdot \mathbf{w}^{\text{T}}\mathbf{w}, \tag{9}$$

where y_n is the label for the n-th sample, $\text{CE}(\cdot, \cdot)$ is the cross-entropy loss and λ controls the scale of the regularizer. The decision boundary given by f_0, especially the scale of $\mathbf{w}$'s components, provides knowledge on individual clusters. For example, the larger w_k is, the more likely that $\mathbf{c}_k$ indicates a malicious pattern.

The other basic classifiers are instantiated as fuzzy decision trees since: (i) They can utilize the results obtained by the fuzzy clustering. (ii) They provide explainable knowledge on the decision boundary. (iii) They are easy and cheap to train and are suitable for the online updating framework.

A fuzzy decision tree is a decision tree with internal nodes replaced by fuzzy logic assertation, a typical fuzzy decision tree indexed by t with one branch:

IF ($\mathbf{x}_{I_{t,1}}$ is $\mathcal{F}_{I_{t,1}}$) **and** ($\mathbf{x}_{I_{t,2}}$ is $\mathcal{F}_{I_{t,2}}$) $\cdots$
and ($\mathbf{x}_{I_{t,L_t}}$ is $\mathcal{F}_{I_{t,L_t}}$)
THEN $\mathbf{x}$ belongs to the malicious class.

processes an instance $\tilde{x}$ by computing the classification score:

$$S_t(\tilde{x}) = \mathcal{O}(\{\mu_{I_{t,i}}(\tilde{\mathbf{x}}_{I_{t,i}})\}_{i=1}^{L_t}),$$

where L_t is the height of the t-th fuzzy decision tree, $I_{t,i}$ is the i-th dimension index in the tree, $\mu_{t,i}$ embeds the i-th clause of the tree by a membership funtion, and $\mathcal{O}$ is a reduce operator such as $\min$ or product to fuzzify the boolean **and**. Each membership function can be a fuzzy version of **is** or **larger/smaller than** as demonstrated by Fig. 6. In particular, each object $\mathcal{F}_{I_{t,l}}$'s center is chosen from the centroids derived from FBPCM.

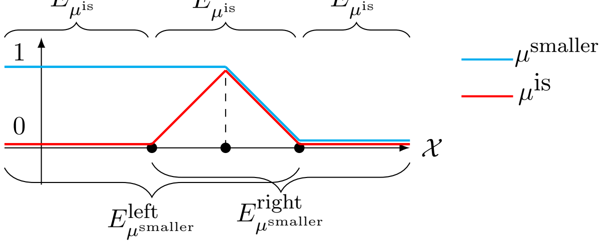


Fig. 6. Fuzzy membership functions in nodes of a fuzzy decision tree.

Algorithm 2 The Adaboost training schedule.

```
Initialize: $\forall n = 1, 2, \cdots, N$, $\gamma_n^{(0)} = 1/N$.
for $t = 0$ to $T$ do
    Train $f_t$ by minimizing: $\sum_{n=1}^{N} \gamma_n^{(t)} \cdot \mathbb{I}[(f_t(\mathbf{x}_n) \neq y_n]$.
    Evaluate $f_t$'s accuracy: $\epsilon_t = \sum_{n=1}^{N} \gamma_n^{(t)} \cdot \mathbb{I}[f_t(\mathbf{x}_n) \neq y_n]$.
    Compute $f_t$'s weight: $\theta_t = 1/2 \log\left(1-\epsilon_t/\epsilon_t\right)$.
    Update samples' weights, $\forall n = 1, 2, \cdots, N$,
        $\hat{\gamma}_n^{(t+1)} = \gamma_n^{(t)} \cdot \exp\{-\theta_t \cdot y_n \cdot f_t(\mathbf{x}_n)\}$,
        $Z = \sum_{n=1}^{N} \hat{\gamma}_n^{(t+1)}$, $\gamma_n^{(t+1)} = \frac{\hat{\gamma}_n^{(t+1)}}{Z}$.
end for
Return: $f(\tilde{\mathbf{x}}) = \text{Sign}\left(\sum_{t=0}^{T} \theta_t \cdot f_t(\tilde{\mathbf{x}})\right)$.
```

As in the boosted fuzzy random forest [46], a collection of T fuzzy decision trees, denoted as $\{f_t\}_{t=1}^{T}$, is generated from the labeled data. At the i-th node within the t-th fuzzy decision tree, the optimal splitting $\mu_{t,i}$ is decided by maximizing the fuzzy version of entropy [49]:

$$E(\mu_{t,i}) = H_2\left(\frac{E^{\text{left}}_{\mu_{t,i}} + E^{\text{right}}_{\mu_{t,i}}}{|\mathbf{X}_{t,i}|}\right),$$

in which $\mathbf{X}_{t,i}$ is the dataset at the current node, $E^{\text{left}}_{\mu_{t,i}}$ is the number of mistaken samples in the left branch of the node w.r.t. $\mu_{t,i}$, i.e., the number of samples that are inconsistent with the dominating class with membership higher than a threshold defined by $\mu_{t,i}$, $E^{\text{right}}_{\mu_{t,i}}$ is defined analogously, and H_2 is the entropy for binary source:

$$H_2(p) = -p \cdot \log_2(p) - (1-p) \cdot \log_2(1-p).$$

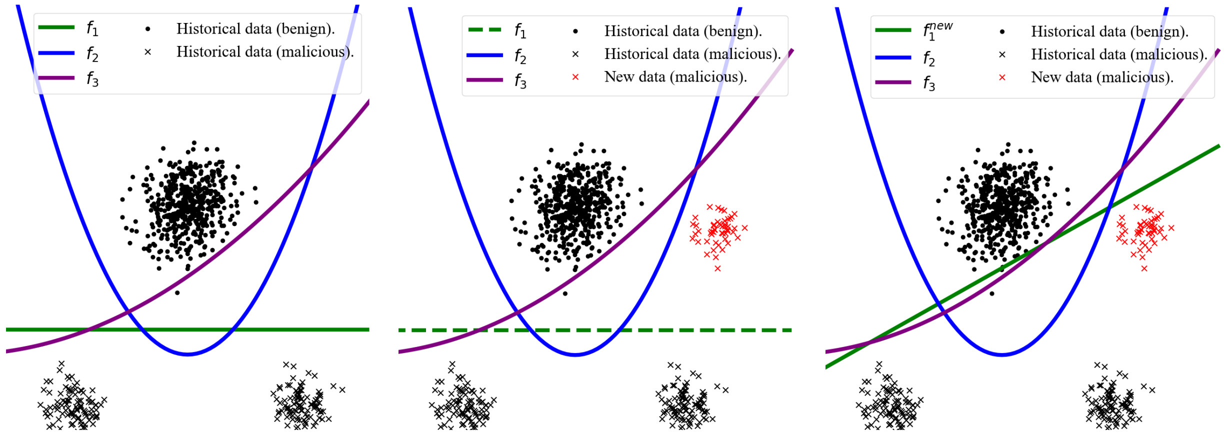
Having determing the optimal split at the current node, $\mathbf{X}_{t,i}$ is splitted along two children nodes:

$$\mathbf{X}_{t,i,\text{left}} = \{\mathbf{x} \in \mathbf{X}_{t,i} : \mu_{t,i}(\mathbf{x}) \in [0, 1/2]\}$$
$$\mathbf{X}_{t,i,\text{right}} = \{\mathbf{x} \in \mathbf{X}_{t,i} : \mu_{t,i}(\mathbf{x}) \in [1/2, 1]\}$$

Once reaching the maximal height, the classification result for the leaf node is the dominating category within its dataset.

2) Ensemble learning for concept drift: All $(T+1)$ basic classifiers are trained following the Adaptive boosting (Adaboost) [50] schedule. Adaboost modifies the weight of all data samples after finishing training each classifier, so the next classifier would focus more on the misclassified samples. Concretely, the ensemble classifier is trained as outlined in Algo. 2.

Having trained all the basic classifiers, we design a scheduling framework that ensembles them and updates their weight to cope with the streaming data. For proper convergence and stable performance, we adhere to the Adaboost framework that assigns weights to both samples and basis classifiers. To emphasize the more up-to-date knowledge carried by recent samples, we modified the conventionalAdaboost process by adopting the weight decaying scheme from reinforcement learning [51], [52], where the arriving order of samples plays a crucial role. Formally, the m-th latest arriving sample has



(a) Classifiers trained from historical data. (b) The concept drift takes place. (c) Classifier update, f_1^{new} replaces f_1.

Fig. 7. Update of the classifier backend against the concept drift.

an extra decaying factor $\delta^{(m-1)}$ with $\delta \in [0, 1]$. To re-evaluate the weight of the t-th basic classifier on the streaming data, we reuse the schedule in Adaboost with the decaying factor for the temporal optimal ensemble.

$$\begin{aligned} \gamma_m^{(t),\dagger} &= \gamma_m^{(t)} \cdot \delta^{(m-1)}, \\ \epsilon_t^\dagger &= \sum_m \gamma_m^{(t),\dagger} \cdot \mathbb{I}[f_t(\mathbf{x}_m) \neq y_m], \\ \theta_t^\dagger &= \frac{1}{2} \log \left({}^{1-\epsilon_t^\dagger}/_{\epsilon_t^\dagger} \right). \end{aligned}$$

If the classifier backend's performance on the arriving data batch drops below a threshold τ then we discard the malfunction weak classifier, i.e., f_t whose $\theta_t^\dagger$ is the smallest, and train a substitute fuzzy decision tree/retrain the linear factor within f_0. This process is visualized in Fig. 7.

Whenever a concept drift is detected, FBPCM is adopted to identify potential new models in each attribute. Then new basic classifiers are tuned by absorbing these emerging centroids as decision boundaries.

3) Computational cost: The cost in tuning the entire classifier is of order $\mathcal{O}(TDKN \cdot 2^H)$, where H is the height of the tree and D is the dimensionality of features since in splitting each node within a fuzzy decision tree, we examine all candidate features and all possible centroids to locate the optimal separation. The cost in maintaining the classifier backend online is at most $\mathcal{O}(DKN \cdot 2^H)$, and the update is conducted in a lazy manner, i.e., the system is updated only when a concept drift is detected.

IV. Experiments and Discussions

A. Data collection

To study the performance of the proposed IDS, we collected the network traffic from a real IoT system to form a new dataset. Our IoT consists of two routers: ASUS AC-86U and D Link 823G, and two cameras: EZVIZ CS-C3C and Mi Home Security Camera 2k as edge devices. Both devices were deployed with publicly available IP addresses and their network traffic was recorded accordingly where we identified four categories of abnormal traffic besides normal traffic: the Man-in-the-Middle attack (`ettercap`), the Reconnaissance attack (`Nmap` and `Sfuzz`), the Distributed Denial-of-Service attack (`thc` and `hping3`), and the Exploit attack (`nmap`). Attacks were conducted from a Ubuntu virtual machine. The eavesdropping lasted for 48 hours from a mirror port using a Win10 server with `Wireshark`. Collected traffic flows in PCAP packages were processed by `Tranalyzer2` to extract features. We also incorporated three public datasets for intrusion detection: KDD'99 [24], UNSW-NB15 [25], and CIC-IDS [53]. Their properties are listed in Table III, where **Ratio** is the number of normal traffic records divided by that of malicious ones. Compared with other datasets, our collection is not artificial and contains the most diversified features.

TABLE III
THE COMPARISON OF STUDIED DATASETS.

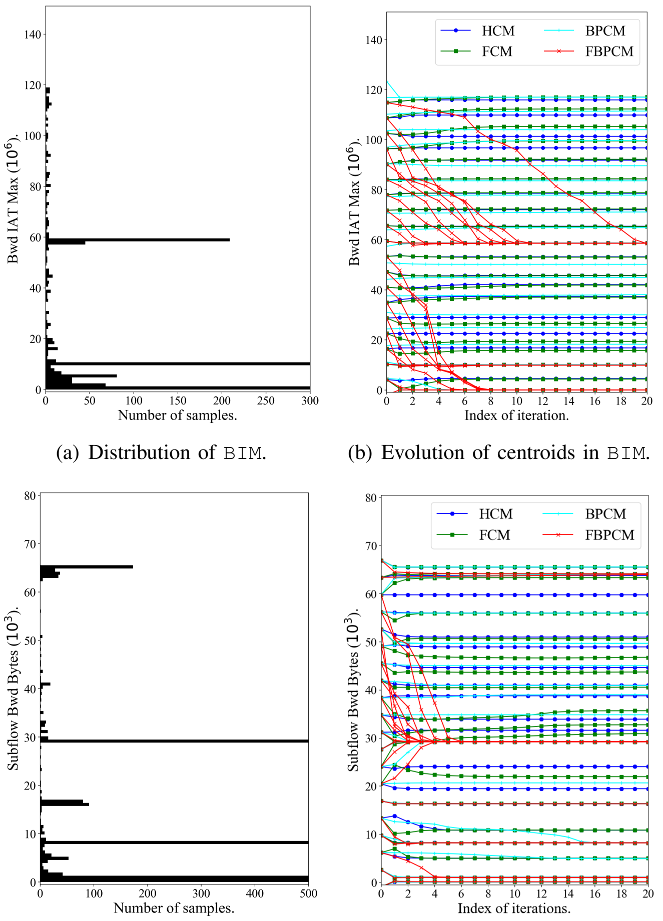
Dataset	#Traffic	#Attack family	#Feature	Ratio	Source
KDD'99	4,898,431	4	42	0.25	Artificial
UNSW-NB15	263,011	9	49	0.47	Artificial
CIC-IDS	2,830,743	4	77	3.79	Artificial
Our dataset	549,810	5	123	0.77	Real

B. The FBPCM module

To evaluate FBPCM's ability to recognize fuzzy clusters and adapt to new patterns, we compared its performance to several ordinary clustering algorithms including HCM, FCM, and BPCM [19].

FBPCM was instantiated with $\alpha = 0.1$ and $\beta = 0.2$, in FCM the fuzziness was set as 2, BPCM was applied with the initial setting in [19]. The distance metric was uniformly the l_2 norm. Along each dimension, $K = 20$ centroids were randomly sampled in the feature space, the clustering algorithm was run until the shift of centroids was smaller than $\epsilon = 0.1$ or the number of iterations reached 20.

Since the statistics of network traffic are usually overwhelmed by noise, it is necessary that the clustering algorithm correctly identifies representative patterns despite outliers. As examples, we selected `Bwd IAT Max` (`BIM`) and



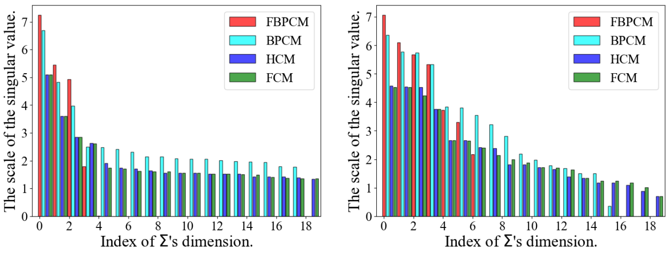
(a) Distribution of `BIM`.

(b) Evolution of centroids in `BIM`.

(c) Distribution of `SBB`.

(d) Evolution of centroids in `SBB`.

Fig. 8. The distribution of two features with severe noise and the routing of centroids for different clustering algorithms.

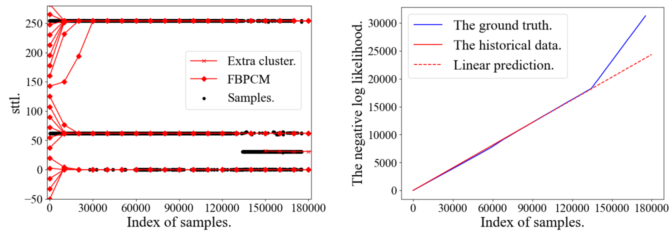


(a) `Bwd IAT Max`.

(b) `Subflow Bwd Bytes`.

Fig. 9. The singular values of the membership matrix for different clustering algorithms, measured in $\log(1+x)$.

`Subflow Bwd Bytes` (`SBB`) in CIC-IDS, which are haunted by severe background noise as shown in Fig. 8 (a)(c). The movement of centroids with the iterations is illustrated in Fig. 8 (b)(d). From which we observed that centroids in FBPCM converged to the representative centers regardless of their initialization. This is because a prior distribution had been introduced for the membership values and the centroids would no longer be captured and misled by local noise.



(a) `sttl`.

(b) The likelihood.

Fig. 10. The distribution of `sttl` and centroids extracted from the first half of data.

The first privilege of this robustness is the easier locating of centroids. As in Fig. 8, the resulting centroids for schemes other than FBPCM remained diversified, so it is necessary to re-examine the membership matrix to differentiate centroids from noise. The second privilege of robustness is the better recognition of the number of representative patterns. We applied SVD to the membership matrices obtained by clustering algorithms on two features. Their singular values are shown in Fig. 9, from which we can unambiguously learn the number of clusters for FBPCM. Meanwhile, the spectrum of other clustering algorithms, with longer tails, is less discriminative regarding the optimal number of clusters. To measure the advantage of FBPCM over other options, we evaluated the averaged fuzzy PBM-index [54] of all clustering algorithms w.r.t. all attributes, the results are shown in Table IV. The fuzzy PBM-index is computed by:

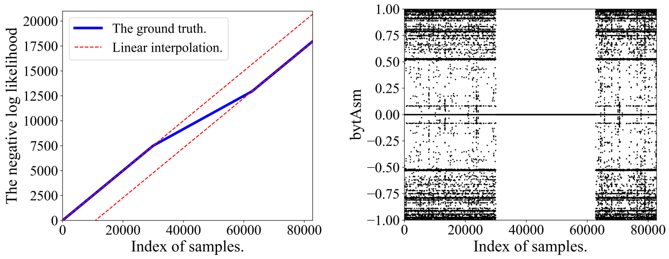
$$I_{\text{fuzzy-PBM}} = \left(\frac{1}{K}\times\frac{E}{J}\times D\right)^2,$$

where K is the number of clusters, E is the deviation of the dataset, D is the maximal distance between centroids, and $J = \sum_{n=1}^{N}\sum_{k=1}^{K} u_{n,k}^{1.5}\|\mathbf{x}_n - \mathbf{c}_k\|$ measures the intrinsic variance within clusters. FBPCM obtained the maximal fuzzy PBM-index since it can usually infer the optimal number of clusters.

TABLE IV
THE AVERAGED FUZZY PBM-INDEX OF CLUSTERING ALGORITHMS.

HCM	FCM	BPCM	FBPCM
5,954	6,412	7,503	13,025

To test the capability of the evidence framework proposed in Sec. III-B3 for emerging patterns, we considered `sttl` in UNSW-NB15 shown in Fig. 10 (a) as an example. Upon receiving the first half of the data stream, FBPCM correctly identified representative patterns. When the second half of data entered the system, the evidence computed according to Eq. (7) took the trend in Fig. 10 (b). Having observed the abnormal growth in the negative log-likelihood, it is likely that a new cluster has formed, then another E-step yielded the result in Fig. 10 (a). The locating of the new centroid is very fast since the prior configuration Eq. (8) naturally contributes the outliers, i.e., those whose membership values to all centroids are low, to the new candidate. The decline speed of the likelihood forms evidence for abnormality. For example, the likelihood of `bytAsm` in our dataset during one slice



(a) The likelihood.

(b) `bytAsm`.

Fig. 11. One DDoS attack reflected by `bytAsm` in one subset of our dataset.

of observations as Fig. 11 (a) indicated that some abnormal event occurs during the intermediate time, which is justified by visualizing all features in Fig. 11 (b). The deviation in likelihood captured one DDoS attack. Notice that as shown by Fig. 11 (a), abnormality does not necessarily increase the likelihood. This is because some intrusions eliminate the normal diversified traffic and result in monotonous statistics. Therefore, the variation of the likelihood slope rather than its sign appears to be an alert for abnormality.

C. The classification module

1) The settings: The performance of the proposed classifier backend was examined in both the offline setting, where it is assumed that abundant data had been collected and the online setting, the more challenging and realistic case. We instantiated the proposed IDS with λ=0.1 in Eq.(9) during the training of f_0. In the online setting, the alert threshold τ was set to 85% of the accuracy on the static training dataset. For comparison, we leveraged several traditional machine learning models as misuse-based classifers: **L**ogistic **R**egression (LR) [55], **D**eep **N**eural **N**etwork (DNN) [56], **S**upport **V**ector **M**achine (SVM) [57], **H**ermit-kernel **S**upport **V**ector **M**achine (SVM) [58], **k**-**N**earest **N**eighbour (kNN) [59], and **R**andom **F**orest [60] (RF). We adopted a DNN with three layers and ReLU activation, an SVM with RBF kernel, k was chosen as 5 in kNN, and $T = 10$ decision trees/fuzzy decision tree with maximal height 10 were incorporated in RF and our scheme. Adam optimizer [61] was utilized uniformly to tune all models.

In the offline setting, the entire dataset was shuffled to ensure that training data and testing data are subject to an identical distribution. In the online setting, the first 50% of the data stream in the original sequence, with possibly preprocessing, was fed into the classifier as the training dataset. The rest 50% data was split into batches of size 40,000. Upon receiving a new batch of data, the classifier's performance was recorded and the classifier was optionally updated with the labels of this new batch.

2) The offline case: In IDS, one metric of interest is the system's complexity including its number of configurable parameters and the time of training/inference. Another metric is the system's accuracy, which is reflected by the precision (Pr), the recall (Rc), and the F-measure (F1) defined as follows:

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}},\ \mathrm{Rc} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},\ \mathrm{F1} = \frac{2}{\frac{1}{\mathrm{Pr}}+\frac{1}{\mathrm{Rc}}},$$

the area under ROC curve (AUC) was also incorporated. All classifiers' complexity and accuracy regarding all datasets are listed in Table V, where **#Param** is averaged across four datasets. Numerically, we conducted the Wilcoxon test on a 20-folded cross-validataion experiment, the results are shown in Table VI. It can be observed that our model statistically outperformed other candidates. The generalization ability of network models as LR or DNN is relatively low due to the multimodel distribution of traffic attributes. The SVM family cannot appropriately identify and cope with the inherent noise and outliers within the data. For RF, the exhaustive search along each dimension for node splitting is very expensive. Despite kNN's high accuracy, its operation time grows linearly with the size of the dataset, so it is inapplicable in online applications. Instead, FBPCM is more robust to noise and outliers and is capable of extracting seminal patterns from raw data. The ensembled fuzzy decision tree backend can conveniently combine the extracted antecedents to yield flexible decision boundaries. As an example, on branch of a fuzzy decision tree extracted for UNSW-NB15 took the form:

$$\begin{aligned} &\textbf{IF}\ (\texttt{dload} \leq 3\times 10^6)\ \textbf{and}\ (\texttt{dttl} \leq 64) \\ &\quad \textbf{and}\ (\texttt{sttl} \leq 63)\ \textbf{and}\ (\texttt{sttl} \geq 60) \\ &\quad \textbf{THEN}\ \text{the flow is malicious.} \end{aligned} \tag{10}$$

Graphically, the dataset before applying each fuzzy logic proposition takes the distribution in Fig. 12. Notice that the splitting points within nodes are centroids from FBPCM and no exhaustive search is involved.

3) The online case: Concept drift commonly takes place in IoT systems, which fact implies that IDS built on the training dataset is going to perform poorly on the testing dataset. For example, a linear classification model trained on 80% data from the ordered/shuffled CIC-IDS achieves (Pr,Rc,F1) of (0.84,0.90,0.87)/(0.91,0.93,0.92) respectively. Therefore, if the deployed IDS is static w.r.t. the data stream, the intrusion detection performance would become unstable, an instance is shown in Fig. 13. To update the IDS under the streaming data, we instantiated our IDS with $T = 10$ fuzzy decision trees and FBPCM as the feature extractor. To simulate the real-world online IoT IDS, we fed the permuted traffic record into the system. Unlike the traditional online learning scenario where the classifier learns from scratch, in IoT, traffic data is abundant. Therefore the IDS was tuned according to the first 50% of traffic records, then the next 50% data arrived and the system conducted classification.

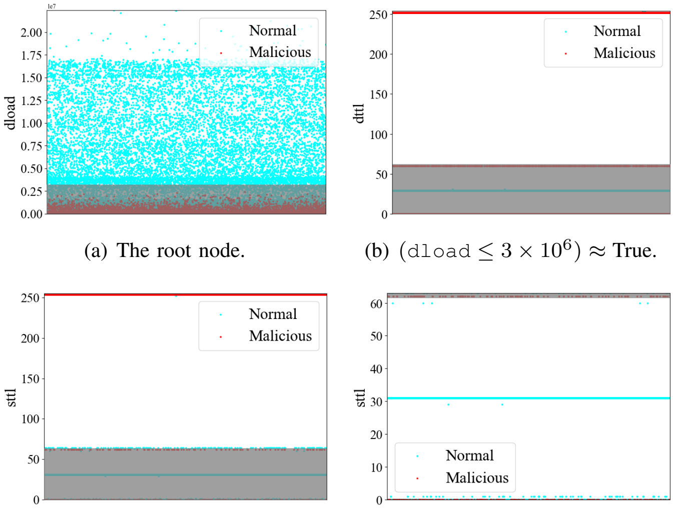
We first studied the model's performance w.r.t. δ, the decay factor of samples' weights in the streaming data, results are shown in Table VII. When δ is large, the IDS is trained with almost the original adaptive boost and no extra attention is given to the emerging pattern. Therefore the discarding and retraining of basic classifiers occur less frequently and the average updating time is shorted. In contrast, a smaller δ results in more focus on recent data. However, a δ too small might suppress the normal behavior of classifiers trained on the historical observations and reduce the accuracy. This could be the reason why that an intermediate assignment of δ usually yielded the optimal performance regarding precision and

TABLE V
THE PERFORMANCE OF MACHINE LEARNING-BASED IDSS ON FOUR IOT TRAFFIC DATASETS. THE OPTIMAL CONFIGURATION IS HIGHLIGHTED.

IDS	**#Param**	**Training time (sec)**	**Inference time (sec)**	**KDD'99**				**UNSW-NB15**				**CIC-IDS**				**Our dataset**			
				Pr	Rc	F1	AUC	Pr	Rc	F1	AUC	Pr	Rc	F1	AUC	Pr	Rc	F1	AUC
LR	156	1.38E+5	2.50E-4	0.947	0.941	0.944	0.947	0.591	0.925	0.721	0.933	0.683	0.777	0.727	0.800	0.777	0.778	0.777	0.802
DNN	702	2.42E+5	3.21E-4	0.950	0.948	0.949	0.955	0.592	0.930	0.724	0.937	0.736	0.746	0.741	0.770	0.780	0.780	0.780	0.804
SVM	156	1.76E+5	5.16E-4	0.815	0.936	0.871	0.944	0.575	0.927	0.710	0.935	0.578	0.691	0.629	0.719	0.766	0.760	0.763	0.781
H-SVM	156	2.09E+5	3.11E-4	0.823	0.945	0.880	0.952	0.593	0.917	0.720	0.926	0.612	0.695	0.651	0.719	0.749	0.782	0.765	0.801
kNN	0	0	5.20E-1	0.973	0.974	0.973	0.977	0.597	0.941	0.730	0.947	0.895	0.920	0.907	0.929	0.786	0.787	0.786	0.805
RF	1,920	2.43E+4	1.28E-3	0.963	0.958	0.960	0.963	0.594	0.934	0.726	0.940	0.854	0.867	0.860	0.879	0.789	0.777	0.783	0.795
Ours	2,310	1.52E+4	3.22E-3	**0.995**	**0.989**	**0.995**	**0.990**	**0.600**	**0.949**	**0.736**	**0.954**	**0.928**	**0.943**	**0.935**	**0.950**	**0.792**	**0.803**	**0.797**	**0.820**

TABLE VI
THE p-VALUE OF REJECTING THE HYPOTHESIS THAT THE COMPARED MODEL IS BETTER THAN OURS.

LR	DNN	SVM	H-SVM	kNN	RF
3.23E-6	2.87E-4	1.88E-4	7.81E-4	1.61E-2	7.11E-3



(a) The root node.

(b) $(\texttt{dload} \leq 3 \times 10^6) \approx$ True.

(c) $(\texttt{dttl} \leq 64) \approx$ True.

(d) $(\texttt{sttl} \leq 63) \approx$ True.

Fig. 12. The distribution of UNSW-NB15 along one branch of the fuzzy decision tree. The shadow areas satisfy the fuzzy decision tree in Eq. (10).

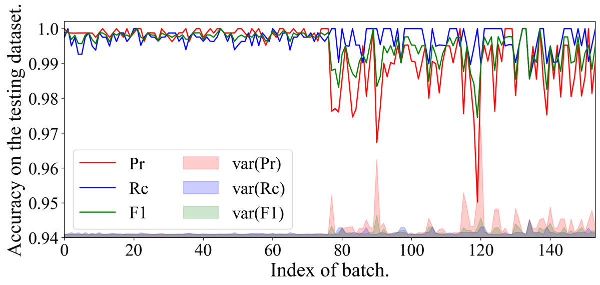


Fig. 13. The performance of one linear classifier on UNSW-NB15 in the online case.

recall. In the following experiments we adopted $\delta = 1 - 2.0$E-5.

For comparison, we also adopted the online version of several misuse-based machine learning IDSs, including DNN-based, SVM-based, and kNN-based. The online version of DNN-based and SVM-based IDSs tunes the model accordingly for each arriving batch, while kNN is inherently an online model. The cumulative F1 score of studied models on four datasets is demonstrated in Fig. 14. Another metric of interest in online IDSs is the cumulative time for model updating, which we recorded and visualize in Fig. 15. We observe that compared with the other three datasets, KDD'99 suffers the least from concept drift, since the variations of online IDSs' performance on KDD'99 were the smallest. Even though all IDS were updated according to the data stream, the performance on new data declined. This is because fine-tuning on the new batches can hardly preserve accuracy and generalization ability as in training using abundant data. In UNSW-NB15 and our dataset, the performance of DNN and SVM dropped significantly, which might be the result of catastrophic forgetting [62], where fine-tuning made the model forget previous knowledge. Instead, our scheme achieved the optimal F1 score with limited updating cost. This is because: (i) Update is done only when the new batch severely deviates from the previous data, so the cumulative time is not necessarily linear in data size. (ii) Training a fuzzy decision tree from extracted centroids is easy compared with tuning parameters with gradient information. Although kNN achieved the best accuracy, its inference time which grows linearly in the data stream's length is unbearable for online IDS.

D. Discussions

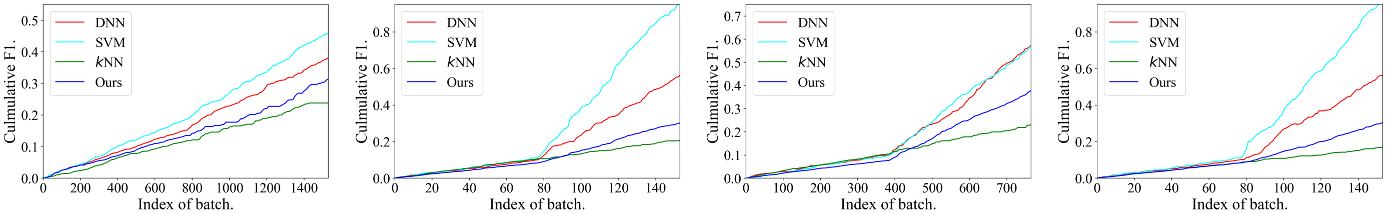
Our IDS, as other misuse-based IDSs, requires labeled traffic datasets for initialization, which is a limitation for some scenarios where labeling is expensive or unavailable. Another drawback of our scheme is the difficulty in determining δ. As what has been shown by Table VII, the optimal δ for different datasets varies. This is because the scale and frequency of concept drifts are different w.r.t. datasets, so δ, which regulates the sensitivity of the proposed IDS, also needs to be modified. The adaptive selection of this hyperparameter remains a challenge for online IDSs. Potential solutions to this challenge include dynamical tuning of δ w.r.t. the system's maintenance and accuracy or running multiple systems with different δ for some time and reserving the optimal one.

V. CONCLUSION

Considering the pervasive noise and distribution shift in the traffic's features in IoT, we design a fuzzy intrusion detection system to protect IoT against malicious communications. To

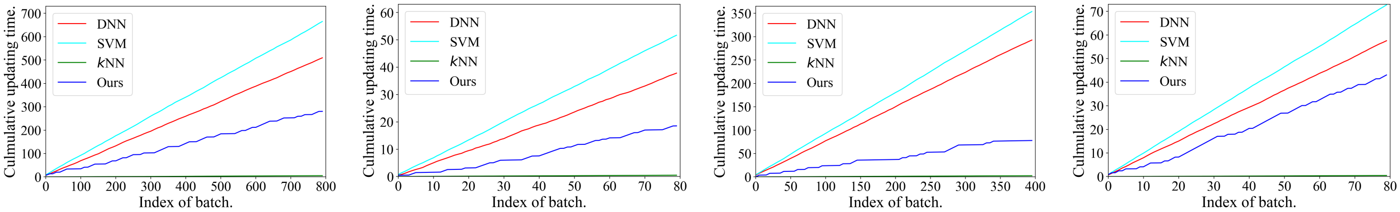
TABLE VII
THE PERFORMANCE OF OUR IDS ON FOUR IOT TRAFFIC DATASETS FOR DIFFERENT δS, THE ONLINE SETTING. THE OPTIMAL CONFIGURATION IS HIGHLIGHTED.

$1-\delta$	KDD'99					UNSW-NB15					CIC-IDS					Our dataset				
	Pr	Rc	F1	AUC	Time	Pr	Rc	F1	AUC	Time	Pr	Rc	F1	AUC	Time	Pr	Rc	F1	AUC	Time
1.0E-6	0.980	0.977	0.978	0.979	8.19E+0	0.569	0.581	0.575	0.610	6.50E+0	0.615	0.573	0.594	0.598	8.98E+0	0.760	0.775	0.767	0.796	5.45E+0
5.0E-6	0.985	0.992	0.988	0.993	1.37E+1	0.571	0.583	0.577	0.614	1.75E+1	0.619	0.580	0.599	0.605	1.35E+1	**0.770**	**0.781**	**0.775**	**0.799**	1.37E+1
2.0E-5	**0.988**	**0.994**	**0.991**	**0.995**	2.93E+1	0.572	0.585	0.578	0.616	1.92E+1	**0.636**	**0.592**	**0.613**	**0.619**	2.69E+1	0.755	0.774	0.765	0.796	1.73E+1
1.0E-4	0.984	0.991	0.988	0.992	4.10E+1	**0.579**	**0.591**	**0.585**	**0.619**	2.78E+1	0.621	0.585	0.603	0.616	4.31E+1	0.749	0.763	0.756	0.787	2.95E+1
5.0E-4	0.970	0.969	0.969	0.973	6.83E+1	0.570	0.583	0.576	0.608	4.87E+1	0.611	0.576	0.593	0.603	1.08E+2	0.735	0.747	0.741	0.773	5.83E+1



(a) KDD'99. (b) UNSW-NB15. (c) CIC-IDS. (d) Our dataset.

Fig. 14. The culmulative F1 score for IDSs in the online setting.



(a) KDD'99. (b) UNSW-NB15. (c) CIC-IDS. (d) Our dataset.

Fig. 15. The culmulative updating time for IDSs in the online setting.

reduce the impact of noise and overfitting, a full Bayesian version of the possibilistic C-means clustering is proposed to extract representative patterns from the observed data. An evidence framework is adopted to determine the number of clusters. To fit the concept drift, an ensemble classifier backend with an adaptively reweighting schedule is designed. The proposed classifier can adaptively discard outdated basic classifiers and achieve the conjugate optimization of the classification accuracy and the update complexity. Experiments on both public datasets and a new dataset collected from real devices justified the privileges of our system and shed light on the application of fuzzy systems on IoT security.

REFERENCES

[1] Mahmoud Ammar, Giovanni Russello, and Bruno Crispo, "Internet of things: A survey on the security of iot frameworks," *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2018.

[2] Chiara Bodei, Stefano Chessa, and Letterio Galletta, "Measuring security in iot communications," *Theoretical Computer Science*, vol. 764, pp. 100–124, 2019.

[3] Pawani Porambage, Mika Ylianttila, Corinna Schmitt, Pardeep Kumar, Andrei Gurtov, and Athanasios V Vasilakos, "The quest for privacy in the internet of things," *IEEE Cloud Computing*, vol. 3, no. 2, pp. 36–45, 2016.

[4] Bharath Sudharsan, John G Breslin, and Muhammad Intizar Ali, "Adaptive strategy to improve the quality of communication for iot edge devices," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*. IEEE, 2020, pp. 1–6.

[5] Mojtaba Eskandari, Zaffar Haider Janjua, Massimo Vecchio, and Fabio Antonelli, "Passban ids: An intelligent anomaly-based intrusion detection system for iot edge devices," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, 2020.

[6] Muhamad Erza Aminanto, Rakyong Choi, Harry Chandra Tanuwidjaja, Paul D Yoo, and Kwangjo Kim, "Deep abstraction and weighted feature selection for wi-fi impersonation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 621–636, 2017.

[7] Mahdi Soltani, Mahdi Jafari Siavoshani, and Amir Hossein Jahangir, "A content-based deep intrusion detection system," *International Journal of Information Security*, pp. 1–16, 2021.

[8] Libo Chen, Yanhao Wang, Quanpu Cai, Yunfan Zhan, Hong Hu, Jiaqi Linghu, Qinsheng Hou, Chao Zhang, Haixin Duan, and Zhi Xue, "Sharing more and checking less: Leveraging common input keywords to detect bugs in embedded systems," in *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[9] Mohamed Amine Ferrag, Leandros Maglaras, Ahmed Ahmim, Makhlouf Derdour, and Helge Janicke, "Rdtids: Rules and decision tree-based intrusion detection system for internet-of-things networks," *Future internet*, vol. 12, no. 3, pp. 44, 2020.

[10] Maryam Yousefnezhad, Javad Hamidzadeh, and Mohammad Aliannejadi, "Ensemble classification for intrusion detection via feature extraction based on deep learning," *Soft Computing*, vol. 25, no. 20, pp. 12667–12683, 2021.

[11] Nour Moustafa, Jill Slay, and Gideon Creech, "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 481–494, 2017.

[12] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[13] Segun I Popoola, Ruth Ande, Bamidele Adebisi, Guan Gui, Mohammad Hammoudeh, and Olamide Jogunola, "Federated deep learning for zero-

day botnet attack detection in iot edge devices," *IEEE Internet of Things Journal*, 2021.

[14] Jobin Wilson, Amit Kumar Meher, Bivin Vinodkumar Bindu, Santanu Chaudhury, Brejesh Lall, Manoj Sharma, and Vishakha Pareek, "Automatically optimized gradient boosting trees for classifying large volume high cardinality data streams under concept drift," in *The NeurIPS'18 Competition*, pp. 317–335. Springer, 2020.

[15] Kashif Ahmad, Majdi Maabreh, Mohamed Ghaly, Khalil Khan, Junaid Qadir, and Ala Al-Fuqaha, "Developing future human-centered smart cities: Critical analysis of smart city security, data management, and ethical challenges," *Computer Science Review*, vol. 43, pp. 100452–100482, 2022.

[16] Ali Bagherinia, Behrooz Minaei-Bidgoli, Mehdi Hosseinzadeh, and Hamid Parvin, "Reliability-based fuzzy clustering ensemble," *Fuzzy Sets and Systems*, vol. 413, pp. 1–28, 2021.

[17] Cheng Kang, Xiang Yu, Shui-Hua Wang, David S Guttery, Hari Mohan Pandey, Yingli Tian, and Yu-Dong Zhang, "A heuristic neural network structure relying on fuzzy logic for images scoring," *IEEE transactions on fuzzy systems*, vol. 29, no. 1, pp. 34–45, 2020.

[18] Mehran Mazandarani and Xiu Li, "Fractional fuzzy inference system: The new generation of fuzzy inference systems," *IEEE Access*, vol. 8, pp. 126066–126082, 2020.

[19] Fangqi Li, Shi-Lin Wang, and Gongshen Liu, "A bayesian possibilistic c-means clustering approach for cervical cancer screening," *Inf. Sci.*, vol. 501, pp. 495–510, 2019.

[20] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," *Proceeding of Network and Distributed Systems Security (NDSS) Symposium 2018*, pp. 1–15.

[21] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He, "A deep learning approach for intrusion detection using recurrent neural networks," *Ieee Access*, vol. 5, pp. 21954–21961, 2017.

[22] Kaiyuan Jiang, Wenya Wang, Aili Wang, and Haibin Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, vol. 8, pp. 32464–32476, 2020.

[23] Weizhi Meng, Elmar Wolfgang Tischhauser, Qingju Wang, Yu Wang, and Jinguang Han, "When intrusion detection meets blockchain technology: a review," *Ieee Access*, vol. 6, pp. 10179–10188, 2018.

[24] H Günes Kayacik, A Nur Zincir-Heywood, and Malcolm I Heywood, "Selecting features for intrusion detection: A feature relevance analysis on kdd 99 intrusion detection datasets," in *Proceedings of the third annual conference on privacy, security and trust*. Citeseer, 2005, vol. 94, pp. 1723–1722.

[25] Nour Moustafa and Jill Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[26] Stefan Burschka and Benoît Dupasquier, "Tranalyzer: Versatile high performance network traffic analyser," in *2016 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2016, pp. 1–8.

[27] S Immaculate Shyla and SS Sujatha, "Cloud security: Lkm and optimal fuzzy system for intrusion detection in cloud environment," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 1626–1642, 2020.

[28] Yingbing Yu and Han Wu, "Anomaly intrusion detection based upon data mining techniques and fuzzy logic," in *2012 IEEE International conference on systems, man, and cybernetics (SMC)*. IEEE, 2012, pp. 514–517.

[29] Salma Elhag, Alberto Fernández, Abdulrahman Altalhi, Saleh Alshomrani, and Francisco Herrera, "A multi-objective evolutionary fuzzy system to obtain a broad and accurate set of solutions in intrusion detection systems," *Soft computing*, vol. 23, no. 4, pp. 1321–1336, 2019.

[30] Mohammad Saniee Abadeh, Hamid Mohamadi, and Jafar Habibi, "Design and analysis of genetic fuzzy systems for intrusion detection in computer networks," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7067–7075, 2011.

[31] Mohammad Reza Mahmoudi, Dumitru Baleanu, Zulkefli Mansor, Bui Anh Tuan, and Kim-Hung Pho, "Fuzzy clustering method to compare the spread rate of covid-19 in the high risks countries," *Chaos, Solitons & Fractals*, vol. 140, pp. 110230, 2020.

[32] Bahrul Ilmi Nasution, Robert Kurniawan, Tiodora Hadumaon Siagian, and Ahmad Fudholi, "Revisiting social vulnerability analysis in indonesia: An optimized spatial fuzzy clustering approach," *International Journal of Disaster Risk Reduction*, vol. 51, pp. 101801, 2020.

[33] Xiaohong Jia, Tao Lei, Xiaogang Du, Shigang Liu, Hongying Meng, and Asoke K Nandi, "Robust self-sparse fuzzy clustering for image segmentation," *IEEE Access*, vol. 8, pp. 146182–146195, 2020.

[34] Liqun Liu, Bing Xu, Xiaoping Zhang, and Xianjun Wu, "An intrusion detection method for internet of things based on suppressed fuzzy clustering," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–7, 2018.

[35] Hafsa Benaddi, Khalil Ibrahimi, and Abderrahim Benslimane, "Improving the intrusion detection system for nsl-kdd dataset based on pca-fuzzy clustering-knn," in *2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)*. IEEE, 2018, pp. 1–6.

[36] Adriana Sayuri Iwashita and João Paulo Papa, "An overview on concept drift learning," *Ieee Access*, vol. 7, pp. 1532–1547, 2018.

[37] Javad Hamidzadeh and Reyhaneh Ghadamyari, "Clustering data stream with uncertainty using belief function theory and fading function," *Soft Computing*, vol. 24, no. 12, pp. 8955–8974, 2020.

[38] Thi Thu Thuy Nguyen, Tien Thanh Nguyen, Alan Wee-Chung Liew, and Shi-Lin Wang, "Variational inference based bayes online classifiers with concept drift adaptation," *Pattern Recognition*, vol. 81, pp. 280–293, 2018.

[39] Roberto Souto Maior de Barros, Silas Garrido T de Carvalho Santos, and Paulo Mauricio Gonçalves Júnior, "A boosting-like online learning ensemble," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 1871–1878.

[40] Federico Maggi, William Robertson, Christopher Kruegel, and Giovanni Vigna, "Protecting a moving target: Addressing web application concept drift," in *International workshop on recent advances in intrusion detection*. Springer, 2009, pp. 21–40.

[41] Chong Di, Fangqi Li, and Shenghong Li, "Sensor deployment for wireless sensor networks: A conjugate learning automata-based energy-efficient approach," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 80–87, 2020.

[42] Giuseppina Andresini, Feargus Pendlebury, Fabio Pierazzi, Corrado Loglisci, Annalisa Appice, and Lorenzo Cavallaro, "Insomnia: Towards concept-drift robustness in network intrusion detection," in *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security (AISec)*. ACM, 2021.

[43] Roberto Jordaney, Kumar Sharad, Santanu K Dash, Zhi Wang, Davide Papini, Ilia Nouretdinov, and Lorenzo Cavallaro, "Transcend: Detecting concept drift in malware classification models," in *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017, pp. 625–642.

[44] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[45] Piero Bonissone, José M Cadenas, M Carmen Garrido, and R Andrés Díaz-Valladares, "A fuzzy random forest," *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 729–747, 2010.

[46] Fangqi Li, Shilin Wang, Alan Wee-Chung Liew, Weiping Ding, and Gong Shen Liu, "Large-scale malicious software classification with fuzzified features and boosted fuzzy random forest," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2020.

[47] Abraham J Wyner, Matthew Olson, Justin Bleich, and David Mease, "Explaining the success of adaboost and random forests as interpolating classifiers," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 1558–1590, 2017.

[48] Todd K Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[49] Mohammed Ramdani, *Système d'induction formelle à base de connaissances imprécises*, Ph.D. thesis, Paris 6, 1994.

[50] Holger Schwenk and Yoshua Bengio, "Training methods for adaptive boosting of neural networks for character recognition," *Advances in neural information processing systems*, vol. 10, pp. 647–653, 1998.

[51] Samir Al-Stouhi and Chandan K Reddy, "Adaptive boosting for transfer learning using dynamic updates," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 60–75.

[52] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[53] Zachariah Pelletier and Munther Abualkibash, "Evaluating the cic ids-2017 dataset using machine learning methods and creating multiple predictive models in the statistical computing language r," *Science*, vol. 5, no. 2, pp. 187–191, 2020.

[54] Malay K. Pakhira, Sanghamitra Bandyopadhyay, and Ujjwal Maulik, "Validity index for crisp and fuzzy clusters," *Pattern Recognition*, vol. 37, no. 3, pp. 487–501, 2004.

[55] Elham Besharati, Marjan Naderan, and Ehsan Namjoo, "Lr-hids: logistic regression host-based intrusion detection system for cloud environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 9, pp. 3669–3692, 2019.

[56] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Pierre-Louis Dubouilh, Ephraim Iorkyase, Christos Tachtatzis, and Robert Atkinson, "Threat analysis of iot networks using artificial neural network intrusion detection system," in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2016, pp. 1–6.

[57] F Eid Heba, Ashraf Darwish, Aboul Ella Hassanien, and Ajith Abraham, "Principle components analysis and support vector machine based intrusion detection system," in *2010 10th international conference on intelligent systems design and applications*. IEEE, 2010, pp. 363–367.

[58] Vahid Hooshmand Moghaddam and Javad Hamidzadeh, "New hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier," *Pattern Recognition*, vol. 60, pp. 921–935, 2016.

[59] Wenchao Li, Ping Yi, Yue Wu, Li Pan, and Jianhua Li, "A new intrusion detection system based on knn classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, vol. 2014, 2014.

[60] Paulo Angelo Alves Resende and André Costa Drummond, "A survey of random forest based methods for intrusion detection systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–36, 2018.

[61] Zijun Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 2018, pp. 1–2.

[62] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan, "Measuring catastrophic forgetting in neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32.

Fang-Qi Li received the M.S. degree in cyber science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2022. He is currently pursuing the Ph.D. degree in cyber science and engineering in the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include the security of machine learning systems and their applications in computer security.

Rui-Jie Zhao received the M.S. degree in cyber science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2021. He is currently pursuing the Ph.D. degree in cyber science and engineering in the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include deep learning, network security, internet of things, etc.

Shi-Lin Wang (Senior Member, IEEE) received the B.Eng. degree in electrical and eletronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2001, and the Ph.D. degree from the Department of Computer Engineering and Information Technology, City University of Hongkong, in 2004. Since 2004, he has been with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, where he is currently a Professor. His research interest interests include image processing and pattern recognition.

Li-Bo Chen received a B.E. degree in computer science and engineering from the PLA Information Engineering University and a Master's degree in Computer Technology from Tsinghua University. He has been with the School of Cyber Science and Engineering in Shanghai Jiao Tong University since 2018. His research mainly involves Software Security, Embedded System Security, and Blockchain Security. He has published more than ten conference and journal papers, including USENIX Security Symposium, IEEE TrustCom, VARA, etc.

Alan Wee-Chung Liew (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electrical and electronic engineering from the University of Auckland, New Zealand, in 1993, and the Ph.D. degree in electronic engineering from the University of Tasmania, Australia, in 1997. He worked as aResearch Fellow and later as a Senior Research Fellow with the Department of Electronic Engineering, City University of Hong Kong. From 2004 to 2007, he was with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, as an Assistant Professor. In 2007, he joined the School of Information and Communication Technology, Griffith University, as a Senior Lecturer, and currently as a Professor. His current research interests include machine learning, pattern recognition, computer vision, medical imaging, and bioinformatics. He serves on the Organizing Committee or the Technical Program Committee for many international conferences. He also serves as an Associate Editor for several journals, such as the IEEE Transactions on Fuzzy Systems.

Weiping Ding (Senior Member, IEEE) received the Ph.D. degree in Computer Science, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2013. From 2014 to 2015, he is a Postdoctoral Researcher at the Brain Research Center, National Chiao Tung University (NCTU), Hsinchu, Taiwan. In 2016, He was a Visiting Scholar at National University of Singapore (NUS), Singapore. From 2017 to 2018, he was a Visiting Professor at University of Technology Sydney (UTS), Ultimo, NSW, Australia. He is currently a professor with the School of Information Science and Technology, Nantong University, Nantong, China. His research interests include deep neural networks, multimodal machine learning, granular data mining, and medical images analysis. He has published more than 150 journal papers, including IEEE T-FS, T-NNLS, T-CYB, T-SMCS, T-BME, T-EVC, T-II, T-ETCI, T-CDS, T-ITS and T-AI. And he has held 20 approved invention patents in total over 35 issued patents. He has co-authored two books. His six authored/co-authored papers have been selected as ESI Highly Cited Papers. Dr. Ding is vigorously involved in editorial activities. He served/serves on the Editorial Advisory Board of Knowledge-Based Systems and Editorial Board of Information Fusion, Engineering Applications of Artificial Intelligence and Applied Soft Computing. He served/serves as an Associate Editor of IEEE Transactions on Neural Network and Learning System, IEEE Transactions on Fuzzy Systems, IEEE/CAA Journal of Automatica Sinica, Information Sciences, Neurocomputing, Swarm and Evolutionary Computation, IEEE Access and Journal of Intelligent & Fuzzy Systems, and Co-Editor-in-Chief of Journal of Artificial Intelligence and System. He is the Leading Guest Editor of Special Issues in several prestigious journals, including IEEE Transactions on Evolutionary Computation, IEEE Transactions on Fuzzy Systems.