Interactive Anomaly Detection in Dynamic Communication Networks

Xuying Meng, Yequan Wang, Suhang Wang, *Member, IEEE*, Di Yao, and Yujun Zhang

***Abstract*— Network flows are the basic components of the Internet. Considering the serious consequences of abnormal flows, it is crucial to provide timely anomaly detection in dynamic communication networks. To obtain accurate anomaly detection results in dynamic networks, supervision from experts is highly demanded. However, to obtain high-quality ground truth of abnormal flows, we suffer from two major problems: (1) *limited labor resources*: experts with the latest domain knowledge are much fewer than the large number of flows; and (2) *dynamic environment*: considering the new abnormal patterns (i.e., new attacks) and continuously changing network structures, it requires timely supervision to adaptively update the parameters. To tackle these problems, we propose HADDN, a novel bandit framework for periodic-updated anomaly detection in dynamic communication networks. We formulate the task as a bandit problem, where by interactions, supervision is offered by human experts to provide the ground truth to a fraction of flows. We construct semi-parametric expected rewards to optimize the estimation of flows' abnormality in limited interactions. Also, we utilize feature-based clusters and structural correlations to make connections between historical flows and new flows to improve both efficiency and accuracy of abnormality estimation. What's more, we provide two implementations for the semi-parametric expected reward of the proposed HADDN with theoretical proof. Experimental evaluations on public datasets demonstrate the substantial improvement of our proposed approaches compared to state-of-art anomaly detection methods.**

***Index Terms*— Anomaly detection, interactive learning, dynamic networks, communication networks, semi-parametric bandits.**

Manuscript received September 23, 2020; revised May 22, 2021 and June 27, 2021; accepted June 30, 2021; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor P. Giaccone. Date of publication July 26, 2021; date of current version December 17, 2021. This work was supported in part by the National Science Foundation of China under Grant 61902382, Grant 61972381, and Grant 62002343; in part by the Research Program of Network Computing Innovation Research Institute under Grant E061010003; in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDC02030500; and in part by the Key Deployment Project of the Chinese Academy of Sciences under Grant KFZD-SW-440. *(Corresponding author: Yujun Zhang.)*

Xuying Meng is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with Purple Mountain Laboratories, Nanjing 211111, China (e-mail: mengxuying@ict.ac.cn).

Yequan Wang and Di Yao are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: tshwangyequan@gmail.com; yaodi@ict.ac.cn).

Suhang Wang is with the College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: szw494@psu.edu).

Yujun Zhang is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the Department of Computer Science and Technology, University of Chinese Academy Sciences, Beijing 100049, China (e-mail: nrcyujun@ict.ac.cn).

Digital Object Identifier 10.1109/TNET.2021.3097137

I. INTRODUCTION

INTERNET is becoming an essential part of our daily life. As necessary components of the Internet, network flows have attracted much attention. Especially, abnormal flows can have great negative impacts on users and Internet-based companies and will lead to catastrophic costs. Thus, to keep the Internet secure, it is crucial to provide timely detection on abnormal flows in dynamic communication networks. Note that, anomaly detection is an umbrella term that aggregates many different tasks, e.g., novelty detection, outlier detection and rare event detection [1]. In this paper, as attacks are employed to jeopardize cyber security and will incur much higher costs than other abnormality types in communication networks, we refer "abnormality" to abnormal flows with attacks (e.g., denial-of-service, infiltration or port scan), and aim to detect these abnormal flows by their dynamic context.

Considering the sparsity of abnormal flows and the abundance of normal flows [1]–[3], it is difficult to learn abnormal patterns from limited abnormal flows, and the imbalanced ratio of abnormal and normal flows makes it hard to learn unbiased classifiers to distinguish the abnormal flows from the vast normal ones. To tackle these problems, most of the existing works are conducted in a fully unsupervised or supervised way [4]–[8]. Unsupervised models [4]–[6] do not require labels but they can apply only to attacks with distinct characteristics (such as DoS attack with bursty traffic flows). For supervised models [7], [8], although they can accurately detect abnormal flows based on the balanced adequate numbers of labeled normal and abnormal flows, most of these works are for static works. With fixed labels, they cannot adapt to the new communication networks with new abnormal patterns. Thus, to adapt to the dynamic environment, we need continuous supervision from experts.

It is very challenging to continuously provide high-quality labels for anomaly detection in dynamic communication networks due to the following two reasons. *First*, labor resources are limited. To find the potential abnormal flows among the vast normal ones, it requires a large number of experts with the latest domain knowledge. Manually offering high-quality labels for all data is labor and time consuming, which is impractical. *Second*, the network is dynamic. New attacks keep arising in the dynamic networks, and can not be detected based on prior labels. Also, the adversarial attacks can add carefully crafted small perturbations to make prior knowledge lose efficacy [9], [10]. There are some initial attempts to reduce limitation from prior labels in a semi-supervised way [11]–[14]. The most relevant work among them is GraphUCB [11] which utilizes multi-armed bandits to conduct anomaly detection with limited feedback from experts. However, it is designed for static networks, where features and structures of each arm do not change with time. In dynamic communication networks,

for flows from the same OD pair (Origin and Destination node pair), both the abnormality degree and the abnormal pattern are changing (i.e., the origin can sometimes send abnormal flows and sometimes send normal ones to the same destination, and different kinds of attacks happen between the same OD pair). Correspondingly, the features and structures are changing, and models on historical labels of these flows can even have negative effects. It is difficult for the existing works to adapt to the new environment especially with limited labor resources.

With the above concerns, we aim to conduct interactive anomaly detection in dynamic communication networks with a small amount of supervision. By interactions, supervision can be provided by human experts and there is high possibility to obtain the ground truth for a small number of flows. The main challenges are (i) how to effectively detect abnormal flows with limited supervision; and (ii) how to adapt to the dynamic communication networks for anomaly detection. In detail, for the first challenge, with limited interactions, there is a dilemma between exploiting historical received labels and exploring flows of new abnormal patterns, where multi-armed bandits are widely used to solve this dilemma. To improve efficiency ad accuracy, we utilize semi-parametric bandits for interactive learning based on the parametric and non-parametric models, respectively. However, since the existing semi-parametric bandits are proposed for recommender systems, and cannot fit the dynamic networks with the new evolving abnormal patterns and continuously changing structures. To adapt to the dynamic networks and tackle the second challenge, we have several observations: in terms of flow features, abnormal flows with similar attacks have similar flow features, which can provide a quick estimation on the abnormality of new flows; and for flow sequences, attackers have high probability to conduct a sequence of abnormal flows towards the target victims, thus origin and destination nodes of each flow can help indicate flows' abnormality. These observations can be utilized to optimize the abnormality estimation results in dynamic networks.

In this paper, we propose a novel Human-in-the-loop framework for Anomaly Detection in Dynamic communication Networks, HADDN, which utilizes contextual bandits to detect abnormal flows in dynamic communication networks with periodic supervision. In detail, we build semi-parametric expected reward to estimate the abnormality based on feature-based clusters and structure characteristic vector for each flow, sharing parameters among flows in each cluster based on the similarities to old flows, and utilizing each flow's origin and destination nodes' historical behaviors to improve the efficiency and accuracy of the abnormality estimation in a dynamic environment. Based on the dynamic expected rewards, we propose to implement two models to adapt to the new communication networks, strategically selecting flows to present to experts and updating the anomaly detection model with the new received labels. Our major contributions are summarized as follows:

- We model the problem of interactive anomaly detection in dynamic communication networks, translating the online stochastic optimization problems into one bandit task;
- We propose HADDN, a periodic update framework for adaptive anomaly detection with the full advantage of limited labels in dynamic communication networks;
- We novelly define the dynamic expected reward by utilizing feature-based clusters and structural correlations to help increase the adaptiveness to the dynamic networks for anomaly detection;
- We introduce two implementations for the semi-parametric expected reward of HADDN with theoretical proof; and
- We demonstrate the substantial improvement of our proposed approaches compared to state-of-art anomaly detection methods.

II. Related Work

A. Anomaly Detection in Communication Networks

It is important to keep communication networks safe and secure, as they are the foundation of various Internet applications. Thus, anomaly detection in communication networks, which aims to detect abnormal flows, is essential. Except those designed for communication networks [2], [3], [15], [16], some existing anomaly detection methods can also work for communication networks, and we categorize them into unsupervised, supervised and semi-supervised ones.

First, for the unsupervised models [4]–[6], these algorithms are designed for some special attacks with distinct characteristics. For example, Pang *et al.* [4] obtain the refined anomaly detection results based on iterative selecting distinguishable features. Eswaran *et al.* [5] conduct anomaly detection based on observations that abnormal flows tend to occur as bursts of activity and will connect parts of the networks which are sparsely connected. As these works treat abnormality in communication networks as general outliers, they can only detect characteristic abnormal flows and can not adapt to different kinds of changing attacks in the dynamic networks.

Second, for the supervised anomaly detection models [7], [8], they can achieve good detection results with balanced numbers of labeled normal and abnormal flows. For example, Zhou *et al.* [7] utilize DNN to train a classifier to detect internet intrusion attacks. However, they are for static networks and labels should keep adequate and balanced, which is unrealistic for dynamic networks.

Third, in order not to be limited to the small numbers of abnormal labels, some semi-supervised detection works [13], [14] assume that anomalies have different characteristics from the normal ones, and fully depend on labeled normal flows. However, they will be subject to the same limitation as unsupervised ones, only work for characteristic attacks. To detect different kinds of attacks, some works utilize *active learning* to present unlabeled data to experts, and incrementally update models based on new labels [11], [12], [17], [18]. For example, Ding *et al.* [11] utilize bandits algorithm to select the most abnormal ones to query experts, treating instances in a community as the same arm. Zha *et al.* [12] utilize reinforcement learning to learn a meta-policy for query selection to maximize the labeled anomalies. However, these works can only work in static networks, and they do not consider the dynamic abnormal patterns and continuously changing structures of each instance, which will greatly reduce the effectiveness of anomaly detection in dynamic networks.

Different from the aforementioned works, we aim to exploit contextual information to detect abnormal flows for various attack types, and investigate semi-parametric bandits to improve the efficiency and accuracy of anomaly detection in dynamic communication networks.

B. Contextual Multi-Armed Bandits

Multi-Armed Bandit (MAB) based models are widely used for active learning, especially in recommender systems [19]. It is usually formulated as a system of many base arms, where the learning agent pulls one arm in every interaction, gets a reward from the environment and tries to maximize the cumulative rewards based on the observed reward. Generally, MAB algorithms can be categorized into two categories, i.e., the classical ones (with non-parametric expected reward) and the contextual ones (with parametric expected reward) [20]. The classical non-contextual MAB algorithms [21], [22] have no contextual features, among which, upper confidence bound (UCB) [21] and Thompson sampling [22] are the most popular ones. For contextual bandits, by utilizing contextual features, they achieve great efficiency especially when the number of arms is large, and have attracted increasing attention in recent years [23]–[25].

To further increase the efficiency and effectiveness, some works utilize clusters to build contextual bandits models with shared parameters [26]–[29] for personalized recommendation. For example, Gentile *et al.* [26] assume users within each group tend to provide similar feedback to different arms, thus they utilize the confidence bound to build user clusters, and share parameters across users in the same cluster. Li *et al.* [27] not only group users but also group items (e.g., arms) based on the similarity of clusterings induced over the users. Although the set of items and users can be changing, they are based on the assumption that users tend to give different feedback on the same arm and the characteristic of each arm do not change, which is completely different to the bandit settings of anomaly detection. For anomaly detection, the abnormal arms should always have a higher possibility to get reward 1, thus different experts should provide similar feedback on the same arm. Also, for each arm, both the abnormality degree and the abnormal patterns are changing, thus the characteristics of arms are changing. In summary, the existing works on the clustering of bandits can not be adopted to anomaly detection in dynamic communication networks.

Some works have utilized contextual MAB to detect anomalies, but most of these works are for outliers [30], [31]. For example, Zhuang *et al.* [30] identify abnormal arms based on the assumption that their expected rewards deviate significantly from most of the other arms. They do not require interactions with human, and obtain rewards based on reward expectation and standard deviation. Except for these general outliers, to adapt to different kinds of abnormalities, some works involve human interactions [11], [12], [32], [33]. However, these works ignore that the characteristics of flows are frequently changing, and cannot adapt to the dynamic communication networks. Thus, we propose to maintain the detection accuracy in dynamic communication networks by sharing the updated value to all flows in the same feature-based cluster and updating semi-parametric expected rewards in each interaction.

III. Problem Statement

We first introduce the notations used in this paper; for convenience, we summarize the main notations used throughout the paper in Table I. All estimated notations are attached with a hat like $\hat{a}$, and the optimal results are attached with an asterisk like a^*. Other notations will be explained in the corresponding sections.

Assuming there are $\{1, \ldots, t, \ldots, T\}$ time periods, the set $\mathcal{D}_t$ is made up of flows appearing in the t-th time period.

TABLE I
TABLE OF MAIN NOTATIONS

Notations	Explanations
d	number of feature dimensions
K	maximum number of interactions for each time
C	maximum number of clusters
t_k	the k-th interaction in the t-th time period
a_{t_k}	the selected flow in t_k
$c(a)$	the cluster that flow a belongs to
$r_{t,a}$	expected reward of flow a in the t-th time period
$\mathbf{x}_{t,a}$	flow a's feature vector in the t-th time period
$\mathbf{z}_{t,a}$	flow a's structural vector in the t-th time period
$\boldsymbol{\theta}_a$	feature weight of flow a
$\boldsymbol{\theta}_{c(a)}$	feature weight of cluster $c(a)$
$\boldsymbol{\rho}$	structure weight
b_a	flow a's optimal bias value
$b_{c(a)}$	cluster $c(a)$'s bias value
$\hat{\boldsymbol{\theta}}_{k,c(a)}$	calculated feature weight of $c(a)$ in t_k
$\boldsymbol{\theta}^*_{c(a)}$	optimal feature weight of cluster $c(a)$
$\hat{\boldsymbol{\rho}}_k$	calculated structure weight in t_k
$\boldsymbol{\rho}^*$	optimal structure weight
$\hat{b}_{k,c(a)}$	calculated bias value of cluster $c(a)$ in t_k
b^*_c	optimal bias value of cluster c
$\hat{r}_{k,a}$	estimated expected reward of flow a in t_k
r^*_k	optimal expected reward of the selected flow in t_k
$\widetilde{r}_k$	received reward in t_k
$\hat{\mathbf{r}}_{t+1}$	estimated abnormality vector of flows in t_k

The feature matrix of all flows across all time periods is denoted as $\mathbf{X}$. The feature matrix of flows on $\mathcal{D}_t$ is $\mathbf{X}_t \in \mathbb{R}^{|\mathcal{D}_t| \times d}$, and $\mathbf{x}_{t,a} \in \mathbb{R}^d$ is the a-th row of $\mathbf{X}_t$. To characterize flow a from various aspects, we construct d-dimension feature vector $\mathbf{x}_{t,a}$ for each flow a with both flow package attributes (e.g., duration) and structural attributes (e.g., counts of flows with the same destination node in the past two seconds) [34]. The OD pair of each flow a is maintained in $\mathcal{P}$, where $\mathcal{P}_a = (i, j)$.

As shown in Fig. 1, in the dynamic networks, both abnormal patterns (i.e., attacks) and network structure change with time. To adapt to the new communication network, we provide K labels in each time period for the periodic update. In detail, in the t-th network, we provide K labels on $\mathcal{D}_t$ to update the model; and before the next periodic update in the $(t+1)$-th network, we can detect anomalies on $\mathcal{D}_{t+1}$ with the updated model.

Although both labeled abnormal and normal flows can improve the performance of anomaly detection, it has been demonstrated that the labeled anomalies provide more help than the normal ones [32] and our experiments in Section VII-D also provide evidence for this observation. Thus we formally define our problem as follows

Given flow feature matrix $\mathbf{X}$ *and their corresponding OD pair set* $\mathcal{P}$*, by interactions with experts, we aim to: (1) in each time period* t*, maximize the number of labeled abnormal flows within* K *interactions, including the new-attack ones; and (2) in each time period* $t+1$*, maximize the detected abnormal flows based on labels received before* $t+1$*.*

IV. Proposed Framework HADDN

In this section, we propose the framework HADDN and translate our two problems into one bandit task. We first provide an overview of the proposed framework.

A. Framework Overview

In dynamic communication networks, new attacks keep arising and can not be characterized by static models. In order

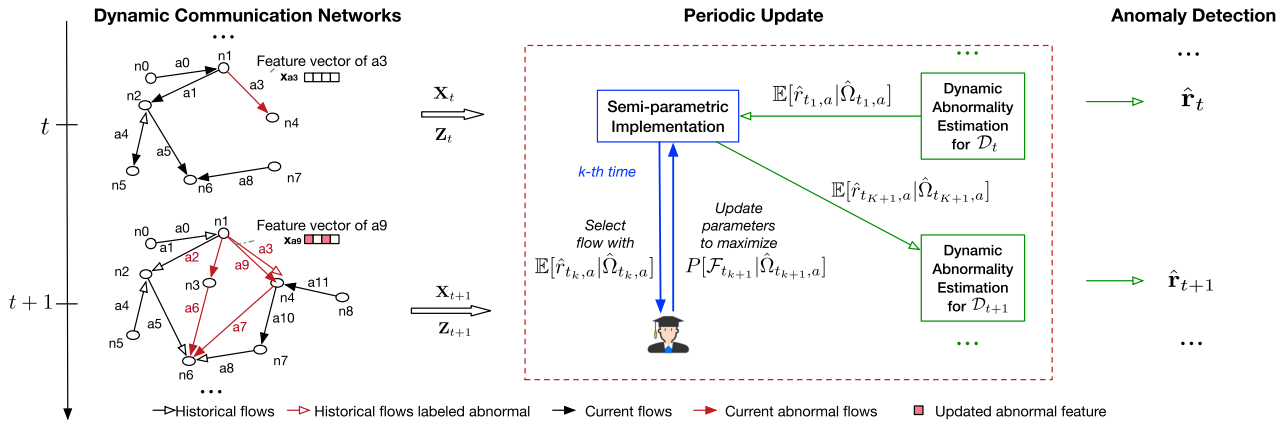


Fig. 1. An illustration of HADDN.

to adapt to new communication networks, we propose a novel Human-in-the-loop framework for Anomaly Detection in Dynamic communication Networks (HADDN). An overview of HADDN is shown in Fig. 1. We divide the framework into two parts, i.e., the periodic update and the anomaly detection. (1) For the periodic update, in each time period, we select flows to present to experts and update the anomaly detection model with the new received labels. The parameter update component helps increase the adaptiveness to the new networks, and the selection policy component will directly decide the number of labeled anomalies. (2) For the anomaly detection, we estimate the abnormality vector $\hat{\mathbf{r}}_{t+1} \in \mathbb{R}^{|\mathcal{D}_{t+1}|}$ for flows on $\mathcal{D}_{t+1}$ by the model learned from the received labels before $t+1$. The dynamic abnormality estimation component will not increase the number of detected anomalies but also help the selection policy to exploit historical labels.

There are many similarities between the task of maximizing the number of labeled abnormal flows and the task of maximizing the number of detected abnormal flows. The difference is that the labeling process needs experts' participation to adapt to the new networks, while the process of detecting abnormal flows does not need experts but a trained detection model. As the dynamic abnormality estimation is a part of selection policy, we translate our two maximization problems into one bandit task, maximizing the labeled abnormal flows in the continuously changing networks with limited labor resources.

B. Contextual Bandits for Dynamic Communication Networks

Bandits are widely used to find the optimal tradeoff between exploring new possibilities and exploiting historical experiences [35]. To adapt to new networks, we utilize contextual bandits to maximize the labeled abnormal flows and improve anomaly detection performance accordingly. In bandits problems, there are multiple arms, where each arm has a probability p_a to get reward 1, and $1-p_a$ to get 0. In each time period, we can try K times, and one arm can be pulled each time. In our scenario, we regard each OD pair as an arm, and flows from some OD pairs have a higher possibility to be abnormal. Flows of each arm can be treated as an outward manifestation of abnormality, and will also be represented as a for simplification. If the k-th presented flow on $\mathcal{D}_t$ is abnormal, we will receive a reward $\widetilde{r}_{t_k} = 1$. To maximize the accumulated rewards in K times interactions, the selection policy will be updated based on the received reward (i.e., label) after each interaction.

To estimate the abnormality (or reward) $r_{t,a}$ of each arm a on $\mathcal{D}_t$, contextual bandits utilize the feature vector $\mathbf{x}_{t,a}$ to increase the convergence speed. The expected reward is

$$r_{t,a} = f(\mathbf{x}_{t,a}, \mathbf{\Omega}_a^*), \tag{1}$$

where $f(\mathbf{x}_{t,a}, \mathbf{\Omega}_a^*)$ is a parametric reward function, $\mathbf{\Omega}_a^*$ is the optimal feature weight for arm a.

We define the history set $\mathcal{F}_{t_k} = \{\mathbf{x}_{\nu,a_{\nu_\tau}}, a_{\nu_\tau}, \widetilde{r}_{\nu_\tau} | (\nu,\tau) \in \Psi_{t_k}\}$, where a_{ν_τ} is the selected flow of the τ-th interaction on $\mathcal{D}_\nu$. The history index Ψ_{t_k} is constructed by the historical interactions before the k-th interaction on $\mathcal{D}_t$, thus $\nu \in \{1,\cdots,t\}$ and $\tau \in \{1,\cdots,k-1\}$. Given $\mathcal{F}_{T_{K+1}}$, the optimal function parameter can be obtained by

$$\mathbf{\Omega}_a^* = \arg\max_{\mathbf{\Omega}_a} P(\mathcal{F}_{T_{K+1}} | \mathbf{\Omega}_a). \tag{2}$$

To maximize the number of labeled abnormal flows, we need a policy that can maximize the cumulative reward. As direct analysis of cumulative reward is not tractable, the cumulative regret $Reg(TK)$ of $T \times K$ interactions is used instead [11], [36], [37]

$$Reg(TK) = \sum_{t=1}^{T}\sum_{k=1}^{K}(w_{t,a_{t_k}^*} r_{t,a_{t_k}^*} - w_{t,a_{t_k}} r_{t,a_{t_k}}), \tag{3}$$

where $a_{t_k}^*$ and a_{t_k} are the optimal arm and the selected arm. To reduce the costs of abnormality, it is instinctive to give expensive flows (i.e., with higher-cost attacks) higher weights, and we provide weights $w_{t,a_{t_k}^*}$ and $w_{t,a_{t_k}}$ for $a_{t_k}^*$ and a_{t_k}. Also, with expensive flows, to increase the detection speed and reduce costs, the scale of time periods should be small. However, updating too often can lead to high labor expenses. The flow weights (or costs) will affect the scale of time periods. Considering the complexity to estimate the real costs (e.g., based on flow locations, attack types and attack scales), for simplification, we set $w_{t,a} = 1$ to be the weight of any arm a in each time period t with the pre-defined time scale.

C. Solution Analysis

Since we cannot obtain the $\mathcal{F}_{T_{K+1}}$ at the beginning, as shown in Fig.1, we update $\hat{\mathbf{\Omega}}_{t_k,a}$ for better abnormality estimation (i.e., expected reward) $\hat{r}_{t_k,a}$ in each interaction.

Algorithm 1 Proposed Framework HADDN

1: **for** $t = 1, \cdots, T$ **do**
2: **for** $k = 1, \cdots, K$ **do**
3: // *Dynamic Abnormality Estimation*
4: **for** $a \in \mathcal{D}_t$ **do**
5: Estimate $\mathbb{E}[\hat{r}_{t_k,a}|\hat{\mathbf{\Omega}}_{t_k,a}]$
6: // *Model Implementation based on* $\hat{r}_{t_k,a}$
7: Select a_{t_k} by the *Section Policy*
8: Receive reward $\widetilde{r}_{t_k}$ from experts
9: *Update parameters* to maximize $P(\mathcal{F}_{t_{k+1}}|\hat{\mathbf{\Omega}}_{t_{k+1},a})$
10: Detect anomalies by $\hat{\mathbf{r}}_{t+1}$ before the $(t+1)$-th update

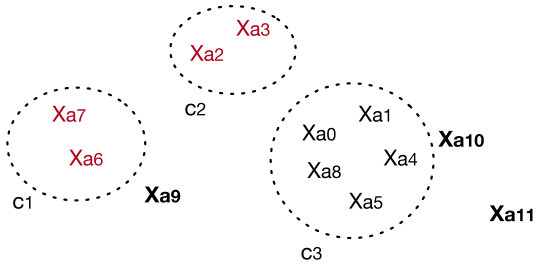


Fig. 2. Feature-based clusters, where the bold denotes features of the flows after t, the unbold are before t, and the red are labeled anomaly.

We describe the process of our bandit-based framework in Algorithm 1. In detail, for the k-th interaction of $\mathcal{D}_t$, we first obtain the abnormality estimation $\mathbb{E}[\hat{r}_{t_k,a}|\hat{\mathbf{\Omega}}_{t_k,a}]$ for each arm a on $\mathcal{D}_t$ from line 4 to 5. To maximize the labeled anomalies in the new network, we implement our bandit model with the estimation $\hat{r}_{t_k,a}$, utilizing our *selection policy* to pursue the tradeoff between exploitation and exploration in line 7, and *updating parameters* for better abnormality estimation with received rewards from line 8 to 9. After K times interactions, we utilize the learned parameters $\hat{\mathbf{\Omega}}_{t_{K+1},a}$ to detect anomalies before the next periodic update, utilizing the abnormality estimation $\hat{\mathbf{r}}_{t+1} = \{\hat{r}_{t+1,a}|a \in \mathcal{D}_{t+1}\} \in \mathbb{R}^{|\mathcal{D}_{t+1}|}$ based on Eq.(1).

With the framework HADDN, there remains two challenges, i.e., (1) how to construct the expected rewards $\mathbb{E}[\hat{r}_{t_k,a}|\hat{\mathbf{\Omega}}_{t_k,a}]$ for accurate abnormality estimation, and (2) based on $\hat{r}_{t_k,a}$, how to accordingly implement the bandit model to maximize the labeled anomalies in the dynamic communication networks. To tackle these two challenges, we will introduce our dynamic abnormality estimation in Section V, and provide two implementation algorithms in Section VI with derived selection policy and parameter update procedures.

V. Dynamic Abnormality Estimation

Traditional contextual bandits construct expected reward fully on the parametric function of features like Eq.(1) [35]. To provide better abnormality estimation for $\mathbb{E}[\hat{r}_{t_k,a}|\hat{\mathbf{\Omega}}_{t_k,a}]$, we define the expected reward in a semi-parametric form, utilizing feature-based clusters and historical flow sequences.

A. Modeling Feature-Based Clusters

In dynamic networks, as shown in Fig. 1 and Fig. 2, new arms keep arising and abnormal patterns of each arm are always changing. For example, flows $a10$ and $a11$ are from new node pairs $(n4, n7)$ and $(n8, n4)$. Also, $a3$ and $a9$ are from $(n1, n4)$ but $\mathbf{x}_{a9}$ is very different to $\mathbf{x}_{a3}$. To improve both efficiency and accuracy of adapting the dynamic communication networks, we need to (1) speed up the abnormality estimation of flows from new arms; and (2) adjust the estimation result for new attacks.

First, as arms have different and changing characteristics in dynamic networks, each arm will have its own parameters $\boldsymbol{\theta}_a$ in Eq. (1). For new arms, in order not to train parameters from scratch and speed up the abnormality estimation process, we utilize cluster-based parameters $\boldsymbol{\theta}_c$. Except for the speedup, the $\boldsymbol{\theta}_c$ will also help adapt to each arm's changing abnormal patterns. In other words, no matter which arm it is and how it changes, flows from these arms are similar if they employ similar attacks. As similar attacks will lead to similar flow features [2], we classify flows into clusters based on features, and flows in the same cluster c share the same $\boldsymbol{\theta}_c$. Thus, we can transform the $(t+1)$-th network in Fig. 1 into Fig. 2. Our contextual reward is defined as

$$r_{t,a} = f(\mathbf{x}_{t,a}, \boldsymbol{\theta}_{c(a)}), \tag{4}$$

where $c(a)$ is a clustering function for flow a.

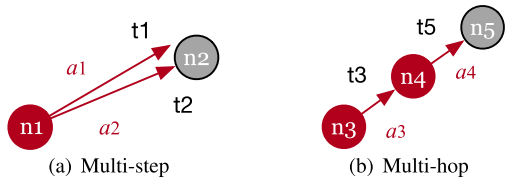
To explain the interactive process, we take Fig. 1 and Fig. 2 as an example. In the t-th network, we construct clusters based on the static features, and obtain the updated parameters $\boldsymbol{\theta}_c$ for each cluster based on K labels, where the reduced parameters will still well estimate the expected rewards for each arm with their personalized $\mathbf{x}_{t,a}$. In the $(t+1)$-th network, to detect anomalies among the new flows (i.e., $a9$, $a10$ and $a11$), we can quickly get the approximate expected rewards by their cluster information. In detail, $a9$ is clustered into $c1$, and $a10$ and $a11$ are clustered into $c3$. As $a9$ is in a cluster most of which are labeled abnormal, and $a11$ is away from the center of its cluster $c3$, thus $a9$ and $a11$ have a high chance to be abnormal and will be presented to experts. In summary, these cluster-based parameters will help speedup the detection process while maintaining the accuracy of the expected reward estimation for each time period.

Second, to adjust the estimation result for new abnormal patterns, we utilize the non-parametric value $b_{c(a_k)}$ as the estimation bias to improve the accuracy.

$$r_{t,a} = f(\mathbf{x}_{t,a}, \boldsymbol{\theta}_{c(a)}) + b_{c(a)}, \tag{5}$$

where $b_{c(a)}$ is different from cluster to cluster. Since clustering results are based on feature similarities [38], irrelevant features will be eliminated before the clustering process. However, new attacks show different dependencies on different features, which may lead to different feature selection results. To well fit the expected reward estimation for new attacks, the adaptive bias can greatly help alleviate the inappropriate features problem for dynamic environment [39].

Any unsupervised feature selection and unsupervised clustering can be adopted in our model. To utilize the mutual effect of feature selection and clustering, we use MCFS (Multi-Cluster Feature Selection) [40] to select features such that the multi-cluster structure of the data can be well preserved. As to clustering, although there are new abnormal patterns in dynamic networks, the number of attacks is relatively steady. For example, there are different kinds of DoS attacks (e.g., ping-of-death, syn flood, smurf), but these DoS attacks have similar features and can be included in one cluster. Thus, here we can fix the number of clusters and perform k-means clustering for simplification. As to the similarity metrics for



(a) Multi-step (b) Multi-hop

Fig. 3. Two kinds of historical flow sequences.

k-means, since different similarity metrics can lead to different clustering results, we utilize Euclidean distances here. That is because some kinds of abnormal patterns (e.g., DoS attacks) can lead increases of not only one feature. Although features' absolute values greatly increase, the feature vectors' relative similarities between the abnormal flows and normal flows (e.g., by cosine distances) may be similar. Also, we will discuss incremental clustering in the next subsection.

B. Modeling Structural Correlations

Although the expected reward of flow a on the same origin and destination node pair (i, j) is not fixed, the historical flows where i or j are involved can still provide guidance for anomaly detection. As shown in Fig. 3, abnormal flows happen in a sequence, i.e., multi-step and multi-hop flow sequences, where the previous detected abnormal flows will increase the abnormal possibility of the upcoming flows [41]. For example, in Fig. 3(a), if $a1$ is detected as an anomaly, there may come another abnormal flows based on the multi-step flow sequences. Thus, the participation of $n1$ can increase $a2$'s probability to be abnormal. Similarly, for the multi-hop flow sequences $a3$ and $a4$ in Fig. 3(b), if $a3$ is a detected anomaly, the participation of $n4$ will also increase $a4$'s probability to be abnormal. In summary, for the origin and destination node i and j of flow a, the frequency of i and j's involvement of abnormal flows can help estimate a's abnormality.

With a small budget to receive feedback, the number of detected abnormal flows is limited. Considering the feature-based clusters in Section V-A, flows in the same cluster share the similar expected reward, which means flows in some of the clusters have a higher possibility to be abnormal. Thus the frequency of i and j's flows in these clusters can help estimate a's possibility to be abnormal. However, we do not know which cluster is abnormal. What's more, there are many flows related to i or j, and these flows may be classified into more than one cluster. It is still difficult to take advantage of the structural correlation information.

In order to learn each cluster's contribution to the abnormality estimation, we construct the structure characteristic vector $\mathbf{z}_a \in \mathbb{R}^C$ for each flow a, where each element $z_a^{(c)}$ denotes the accumulated number of its historical structural correlated flows in the cluster c. We take $a9$ in Fig. 1 as an example. Without considering the new emerged flows, the origin $n1$ of $a9$ has two flows in cluster $c2$ and two in $c3$, and the destination $n4$ has one in $c1$ and one in $c2$, thus we have $\mathbf{z}_{a9} = [1, 3, 2]$ based on flows of $\mathcal{D}_t$. The involvement of $c1$ and $c2$ (i.e., clusters with many labeled abnormal nodes) would contribute to $a9$'s abnormality estimation. To utilize the clustering statistic information $\mathbf{z}_{t,a}$ based on the historical structure correlations, the expected reward is

$$r_{t,a} = f(\mathbf{x}_{t,a}, \boldsymbol{\theta}_{c(a)}) + g(\mathbf{z}_{t,a}, \boldsymbol{\rho}) + b_{c(a)}, \tag{6}$$

where $\boldsymbol{\rho}$ is a parameter vector shared to all arms, as it evaluates the abnormality of each cluster and will not be affected by each individual arm. Note that the initial motivation of clustering is to utilize the arms' current attacks, but structural correlations can only help detect anomalies based on historical abnormal flows and are not related to exact abnormal patterns. Thus, structural correlations should not join the clustering process.

In the $(t+1)$-th network, if new attacks make the corresponding flows largely deviate the existing feature-based clusters, the fixed number of clusters is not suitable to provide an accurate expected reward estimation for these flows any more. Besides, it will affect the parameter optimization of the existing clusters and further decrease the detection performance. To tackle these problems, we can utilize incremental clustering [42], [43] for more precise and adaptive clustering results. Although $g(\cdot)$ can be defined unaffected by the updated dimensions of $\mathbf{z}_{t,a}$ and $\boldsymbol{\rho}$, the corresponding cluster-related parameters $\boldsymbol{\theta}_c$ and b_c for the new cluster should be trained from scratch, which will decrease the model efficiency to some degree. There is a tradeoff between accuracy and efficiency.

VI. Semi-Parametric Implementation

To achieve accurate and efficient anomaly detection results for the dynamic communication networks, we utilize feature-based clusters and structural correlations to define the reward $r_{t,a}$ in a semi-parametric form by Eq. (6). Although the $\mathbb{E}[\hat{r}_{t_k,a}|\hat{\boldsymbol{\Omega}}_{t_k,a}]$ can help estimate the abnormality with exploitation of historical labels, it is still important to explore new possibility, especially for new attacks. In order to meet the balance between exploration and exploitation in dynamic communication networks, we provide two semi-parametric linear implementations based on two widely-used bandit strategies, i.e., Upper Confidence Bound (UCB) [21] and Thompson Sampling (TS) [22], to select flows to present to experts and update parameters to maximize $P(\mathcal{F}_{t_{k+1}}|\hat{\boldsymbol{\Omega}}_{t_{k+1},a})$ accordingly.

We first define the linear expected reward. There are plenty of parametric functions for $f(\cdot)$ and $g(\cdot)$, among which, linear function is the most widely used. In this work, we utilize linear function for the parametric part of the expected reward in the t-th interaction, i.e., $f(\mathbf{x}_a, \boldsymbol{\theta}_{c(a)}) = \mathbf{x}_a^T \boldsymbol{\theta}_{c(a)}$ and $g(\mathbf{z}_a, \boldsymbol{\rho}) = \mathbf{z}_a^T \boldsymbol{\rho}$, where $\boldsymbol{\theta}_{c(a)} \in \mathbb{R}^d$ and $\boldsymbol{\rho} \in \mathbb{R}^C$. The expected reward is

$$r_a = \mathbf{x}_a^T \boldsymbol{\theta}_{c(a)} + \mathbf{z}_a^T \boldsymbol{\rho} + b_{c(a)}, \tag{7}$$

Note that, as algorithms are the same for each time period t, we utilize k, $\mathbf{x}_{a_k}$, $\mathbf{z}_{a_k}$ to denote t_k, $\mathbf{x}_{t,a_{t_k}}$ and $\mathbf{z}_{t,a_{t_k}}$ in the following sections for notation simplification.

A. UCB-Based Strategy

Upper Confidence Bound is widely used to explore new possibilities based on expected rewards. As an UCB-based algorithm, to minimize accumulated regret $Reg(TK)$, there are two steps in the k-th interaction of each $\mathcal{D}_t$:

- *Selection Policy:* Calculate the upper confidence bounds (UCBs) $\mathbf{U}_k$ for the expected rewards, and select the arm a_k with highest UCB to present to experts; and
- *Parameter Update:* Update parameters to maximize $P(\mathcal{F}_{t_{k+1}}|\hat{\boldsymbol{\Omega}}_{t_{k+1},a})$ with the new received reward $\widetilde{r}_k$ for improve the estimation of both expected rewards and UCBs in the $(k+1)$-th interaction.

To provide the implementable algorithm, we first derive the UCBs based on the proposed expected reward in Eq. (6),

and will introduce the corresponding parameter update in the following subsections.

1) Selection Policy: As the non-parametric part $b_{c(a_k)}$ is the bias of the received reward and the expected reward, we define $\hat{b}_{k,c(a_k)} = \bar{r}_{k,c(a_k)} - \bar{f}(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{c(a_k)}) - \bar{g}(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}})$, where $\bar{r}_{k,c(a_k)}$ is the average of received rewards of arms in cluster $c(a_k)$, $\bar{f}(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{k,c(a_k)}) + \bar{g}(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}}_k)$ represents the average of expected rewards of arms in $c(a_k)$. Let $o_{c(a_k)}$ denote the number of received rewards of arms in $c(a_k)$ before k-th interaction, we can define $\bar{r}_{k,c(a_k)} = \frac{1}{o_{c(a_k)}} \sum_{\tau=1}^{o_{c(a)}} \widetilde{r}_\tau$, $\bar{f}(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{k,c(a_k)}) = \frac{1}{o_{c(a_k)}} \sum_{\tau=1}^{o_{c(a_k)}} \mathbf{x}_{\tau,a_\tau}^T \hat{\boldsymbol{\theta}}_{k,c(a_k)}$ and $\bar{g}(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}}_k) = \frac{1}{o_{c(a_k)}} \sum_{\tau=1}^{o_{c(a_k)}} \mathbf{z}_{\tau,a_\tau}^T \hat{\boldsymbol{\rho}}_k$. Accordingly, we have the average optimal expected reward $\bar{r}^*_{k,c(a_k)} = \bar{f}(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*_{c(a_k)}) + \bar{g}(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*) + b^*_{c(a_k)}$ for the optimal expected reward $r^*_{k,a}$ of each a in cluster $c(a_k)$ based on the optimal parameters $b^*_{c(a_k)}$, $\boldsymbol{\theta}^*_{c(a_k)}$ and $\boldsymbol{\rho}^*$. Next, to derive the upper bound of the expected reward in Theorem 1, we divide it into three sub-equations, and will first introduce Lemma 1, 2 and 3 to derive the three sub-equations.

Lemma 1: Let $\alpha_k = R\sqrt{-2ln(\delta/2)/o_{c(a_k)}}$, *where* R *is a positive scalar. If the received reward* $\widetilde{r}_k$ *has a* R*-sub-Gaussian tail* $\eta_k = \widetilde{r}_k - r^*_{k,a_k}$, *i.e.,* $\mathbb{E}(e^{\mu\eta_k}) \leq e^{R^2\mu^2/2}$, *then, with probability at least* $1-\delta$, *we have*

$$|\bar{r}_{k,c(a_k)} - \bar{f}(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*_{c(a_k)}) - \bar{g}(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*) - b^*_{c(a_k)}| < \alpha_k. \quad (8)$$

Proof: Since η_k is a zero-mean Gaussian noise lying in $[-R, R]$, by Hoeffding inequality, we have

$$\begin{aligned} &P(|\bar{r}_{k,c(a_k)} - \bar{f}(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*_{c(a_k)}) - \bar{g}(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*) - b^*_{c(a_k)}| < \alpha_k) \\ &= P(|\bar{r}_{k,c(a_k)} - \bar{r}^*_{k,c(a_k)}| \leq \alpha_k) = P(|\frac{1}{o_{c(a_k)}} \sum_{\tau=1}^{o_{c(a_k)}} \eta_\tau| \leq \alpha_k) \\ &\geq 1 - 2\exp(-\frac{2o^2_{c(a_k)}\alpha^2_k}{\sum_{o_{c(a_k)}}(R-(-R))^2}) = 1-\delta. \end{aligned} \quad (9)$$

which completes the proof. □

Lemma 2: With probability at least $1-\delta/T$, *we have*

$$\begin{aligned} |f(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{k,c(a_k)}) - \bar{f}(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{k,c(a_k)}) \\ - f(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*_{c(a_k)}) + \bar{f}(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*_{c(a_k)})| < \beta_k \end{aligned} \quad (10)$$

where $\beta_k = (R_\beta + 1)\sqrt{\Delta\mathbf{x}^T_{k,a_k}\Delta\mathbf{A}^{-1}_{k-1}\Delta\mathbf{x}_{a_k}} = (R_\beta + 1)\|\Delta\mathbf{x}^T_{k,a_k}\|_{\Delta\mathbf{A}^{-1}_{k-1}}$ *with* $R_\beta = \sqrt{\frac{1}{2}\ln\frac{2KC}{\delta}}$, $\Delta\mathbf{x}_{a_k} = \mathbf{x}_{a_k} - \frac{1}{o_{c(a_k)}}\sum_{\tau=1}^{o_{c(a_k)}}\mathbf{x}_{\tau,a_\tau}$ *and* $\Delta\mathbf{A}_{k-1} = \mathbf{I}_d + \sum_{\tau=1}^{o_{c(a_k)}}\Delta\mathbf{x}_{\tau,a_\tau}\Delta\mathbf{x}^T_{\tau,a_\tau}$.

Proof: As $f(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{k,c(a_k)}) - \bar{f}(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{k,c(a_k)}) - f(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*_{c(a_k)}) + \bar{f}(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*{}_{c(a_k)})$ can be regarded as $(\Delta\mathbf{x}_{a_k})\hat{\boldsymbol{\theta}}_{k,c(a_k)} - (\Delta\mathbf{x}_{a_k})\boldsymbol{\theta}^*_{c(a_k)}$, this lemma can be proven based on the Lemma 1 of [35], □

Lemma 3: With probability at least $1-\delta/K$, *we have*

$$|g(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}}_k) - \bar{g}(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}}_k) + \bar{g}(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*) - g(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*)| < \gamma_k,$$

where $\gamma_k = (R_\gamma+1)\|\Delta\mathbf{z}^T_{k,a_k}\|_{\Delta\mathbf{B}^{-1}_{k-1}}$, *and* $R_\gamma = \sqrt{\frac{1}{2}\ln\frac{2KN}{\delta}}$. *Let* o_k *denote the number of received arms before the* k*-th interaction, then* $\Delta\mathbf{z}_{a_k} = \mathbf{z}_{a_k} - \frac{1}{o_k}\sum_{\tau=1}^{o_k}\mathbf{z}_{\tau,a_\tau}$, $\Delta\mathbf{B}_{k-1} = \mathbf{I}_C + \sum_{\tau=1}^{o_k}\Delta\mathbf{z}^T_{\tau,a_\tau}\Delta\mathbf{z}_{\tau,a_\tau}$.

Proof: As $g(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}}_k) - \bar{g}(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}}_k) + \bar{g}(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*) - g(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*)$ can be regarded as $(\Delta\mathbf{z}_{a_k})\hat{\boldsymbol{\rho}} - (\Delta\mathbf{z}_{a_k})\boldsymbol{\rho}^*$, this lemma can be proven in the similar way of Lemma 2. □

Theorem 1: With probability at least $1-\delta$, *we have*:

$$|\hat{r}_{k,a_k} - r^*_k| \leq \alpha_k + \beta_k + \gamma_k \quad (11)$$

Proof: Based on Lemma 1, 2 and 3, we arrive at

$$\begin{aligned} |\hat{r}_{k,a_k} - r^*_k| &= |\bar{r}_{k,c(a_k)} - \bar{f}(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{k,c(a_k)}) - \bar{g}(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}}_k) \\ &\quad + f(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{k,c(a_k)}) + g(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}}_k) \\ &\quad - f(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*_{c(a_k)}) - g(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*) - b^*_{c(a_k)}| \\ &= |\bar{r}_{k,c(a_k)} - \bar{f}(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*_{c(a_k)}) - \bar{g}(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*) - b^*_{c(a_k)} \\ &\quad + f(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{k,c(a_k)}) - \bar{f}(\mathbf{x}_{a_k}, \hat{\boldsymbol{\theta}}_{k,c(a_k)}) \\ &\quad - f(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*_{c(a_k)}) + \bar{f}(\mathbf{x}_{a_k}, \boldsymbol{\theta}^*_{c(a_k)}) \\ &\quad + g(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}}_k) - \bar{g}(\mathbf{z}_{a_k}, \hat{\boldsymbol{\rho}}_k) \\ &\quad + \bar{g}(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*) - g(\mathbf{z}_{a_k}, \boldsymbol{\rho}^*)| \\ &\leq \alpha_k + \beta_k + \gamma_k \end{aligned} \quad (12)$$

which completes the proof. □

Based on Theorem 1, the upper confidence bound of expected reward for each arm a in the k-th interaction is

$$\mathbf{U}_{k,a} = f(\mathbf{x}_a, \hat{\boldsymbol{\theta}}_{k,c(a)}) + g(\mathbf{z}_a, \hat{\boldsymbol{\rho}}_k) + \hat{b}_{k,c(a)} + \alpha_k + \beta_k + \gamma_k. \quad (13)$$

2) Parameter Update: In order to minimize the accumulated regret $Reg(TK)$, we need to learn the optimal coefficients to estimate the expected rewards. Thus, we utilize ridge regression [35] to learn the $\boldsymbol{\theta}^*_c$ and $\boldsymbol{\rho}^*$ that can best fit all receive rewards $\widetilde{r}_k$ by

$$\sum_k^K(\widetilde{r}_k - \mathbf{x}^T_{a_k}\boldsymbol{\theta}^*_{c(a_k)} - \mathbf{z}^T_{a_k}\boldsymbol{\rho}^*)^2 + \sum_c \|\boldsymbol{\theta}^*_c\|_2 + \|\boldsymbol{\rho}^*\|_2, \quad (14)$$

and b^*_c can be obtained based on the $\boldsymbol{\theta}^*_c$, $\boldsymbol{\rho}^*$ and Eq. (7).

In each interaction, the close-form estimation of the coefficients can be achieved by setting the derivative of Eq. (14) with respect to $\hat{\boldsymbol{\rho}}_k$ and $\hat{\boldsymbol{\theta}}_{k,c(a_k)}$ to be zero, and we have

$$\begin{aligned} \hat{\boldsymbol{\rho}}_k &= \mathbf{P}^{-1}_k\mathbf{Q}_k, \\ \hat{\boldsymbol{\theta}}_{k,c(a_k)} &= \mathbf{A}^{-1}_{k,c(a_k)}(\mathbf{B}_{k,c(a_k)} - \mathbf{C}_{k,c(a_k)}\hat{\boldsymbol{\rho}}_k), \end{aligned} \quad (15)$$

where we define $\mathbf{I}_d \in \mathbb{R}^{d\times d}$ and $\mathbf{I}_C \in \mathbb{R}^{C\times C}$ to be two identity matrices, and

$$\begin{aligned} \mathbf{P}_k &= \mathbf{I}_C + \sum_{\tau=1}^k \mathbf{z}_{\tau,a_\tau}\mathbf{z}^T_{\tau,a_\tau}, \\ \mathbf{Q}_k &= \sum_{\tau=1}^k \mathbf{z}_{\tau,a_\tau}(\widetilde{r}_\tau - \mathbf{x}^T_{\tau,a_\tau}\hat{\boldsymbol{\theta}}_{\tau,c(a_\tau)}), \\ \mathbf{A}_{k,c(a_k)} &= \mathbf{I}_d + \sum_{\tau=1}^k \mathbf{x}_{\tau,a_\tau}\mathbf{x}^T_{\tau,a_\tau}, \\ \mathbf{B}_{k,c(a_k)} &= \sum_{\tau=1}^k \mathbf{x}_{\tau,a_\tau}\widetilde{r}_\tau, \quad \mathbf{C}_{k,c(a_k)} = \sum_{\tau=1}^k \mathbf{x}_{\tau,a_\tau}\mathbf{z}^T_{\tau,a_\tau}. \end{aligned} \quad (16)$$

3) UCB-Based Online Learning Algorithm: With the UCB-based strategy, we exploit historical feedback based on the expected reward in Eq. (7) and *explore* new possibility by the upper bound in Eq. (13). We summarize the UCB-based online updating Algorithm 2. In each interaction, each flow is evaluated by the UCB with Eq. (13). After we present the arm with the highest UCB, we receive labels from experts, and parameters can be updated by Eq. (15) based on the labels. Taking the process in the t-th network as an example, we first initialize the parameters and clusters from line 1 to 5, and conduct K times interactions from line 7 to 26. Specially, in each interaction, we estimate the UCBs for each arm in line 10, present the arm with the highest $\mathbf{U}_{k,a}$ and receive experts' feedback in line 11. The parameter update is from line 13 to 25. As there are two coefficients $\boldsymbol{\theta}_k$ and $\boldsymbol{\rho}_k$, the update of $\mathbf{P}$ and $\mathbf{Q}$ will be mutually affected by the $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ in Eq. (15), we adopt the similar procedures as [36] in line 13 to 19. With the learned parameters, we detect anomalies with $\hat{\mathbf{r}}_{t+1}$ based on Eq.(6).

Based on Theorem 2, we bound the regret of UCB_HADDN to $\tilde{O}(\sqrt{TKd})$, which is common in contextual bandits [21]. However, different from the existing algorithms, we separate these interactions into T time intervals, thus we can keep the freshness of labels to adapt to new communication networks and achieve better anomaly detection performance with feature-based clusters and structural correlations.

Theorem 2: With probability at least $1-\delta$, the regret of the algorithm is

$$O(\sqrt{CTK\ln\delta}+\sqrt{TKd\ln^3(CTK\ln(TK)/\delta)} + \sqrt{TKd\ln^3(NTK\ln(TK)/\delta)}) \quad (17)$$

Proof: With $\alpha_k = R\sqrt{-2ln(\delta/2)/o_{c(a_k)}}$, $\beta_k = (R_\beta+1)\|\Delta\mathbf{x}^T_{k,a_k}\|_{\Delta\mathbf{A}^{-1}_{k-1}}$ and $\gamma_k = (R_\gamma+1)\|\Delta\mathbf{z}^T_{k,a_k}\|_{\Delta\mathbf{B}^{-1}_{k-1}}$, we have

$$\sum_{t=1}^{T}\sum_{k=1}^{K}\alpha_k \le O(\sqrt{\ln(\delta)}\sum_t\sum_k\frac{1}{\sqrt{o_{c(a_k)}}}) \le O(\sqrt{CTK\ln\delta}). \quad (18)$$

Then the regret can be proved by the Theorem 1 in [35] with

$$\sum_{t=1}^{T}\sum_{k=1}^{K}\beta_k \le O(\sqrt{TKd\ln^3(CTK\ln(TK)/\delta)}$$
$$\sum_{t=1}^{T}\sum_{k=1}^{K}\gamma_k \le O(\sqrt{TKd\ln^3(NTK\ln(TK)/\delta)}. \quad (19)$$

which completes the proof. □

Algorithm 2 UCB_HADDN

Input: $\mathbf{X}$, $\mathcal{P}$, $\mathcal{D}_t$, K, C, d, R, δ

1: // Initialization
2: Construct C clusters with $c(a)$ based on $\mathbf{X}$, and construct $\mathbf{Z}$ by $\mathcal{P}$
3: $\mathbf{P}_0 \leftarrow \mathbf{I}_C$, $\mathbf{Q}_0 \leftarrow \mathbf{0}_C$, $o_0 \leftarrow 0$, $R_\beta \leftarrow \sqrt{\frac{1}{2}\ln\frac{2TC}{\delta}}$, $R_\gamma \leftarrow \sqrt{\frac{1}{2}\ln\frac{2TN}{\delta}}$, $\alpha_0 \leftarrow R\sqrt{-2ln(\delta/2)}$, $\beta_0 \leftarrow \alpha_0$, $\gamma_0 \leftarrow \alpha_0$
4: **for** $c \in C$ **do**
5: $o_c \leftarrow 0$, $\mathbf{A}_{0,c} \leftarrow \mathbf{I}_d$, $\mathbf{C}_{0,c} \leftarrow \mathbf{0}_{d\times C}$, $\mathbf{B}_{0,c} \leftarrow \mathbf{0}_{d\times 1}$
6: // In the t-th communication network
7: **for** k $= 1, \ldots,$ K **do**
8: // Select the presented flow based on $\hat{r}_{t,a}$
9: **for** $a \in \mathcal{D}_t$ **do**
10: $\mathbf{U}_{k,a} \leftarrow \mathbf{x}_a^T\hat{\boldsymbol{\theta}}_{k,c(a)} + \mathbf{z}_a^T\hat{\boldsymbol{\rho}}_k + \hat{b}_{k,c(a)} + \alpha_{k-1} + \beta_{k-1} + \gamma_{k-1}$
11: Choose $a_k = \arg\max_{a\in\mathcal{D}_k}\mathbf{U}_{k,a}$ and get the feedback $\widetilde{r}_k$ from experts
12: // Parameter update
13: $\mathbf{P}_k \leftarrow \mathbf{P}_{k-1} + \mathbf{C}^T_{k-1,c(a_k)}\mathbf{A}^{-1}_{k-1,c(a_k)}\mathbf{C}_{k-1,c(a_k)}$
14: $\mathbf{Q}_k \leftarrow \mathbf{Q}_{k-1} + \mathbf{C}^T_{k-1,c(a_k)}\mathbf{A}^{-1}_{k-1,c(a_k)}\mathbf{B}_{k-1,c(a_k)}$
15: $\mathbf{A}_{k,c(a_k)} \leftarrow \mathbf{A}_{k-1,c(a_k)} + \mathbf{x}_{a_k}\mathbf{x}^T_{a_k}$
16: $\mathbf{C}_{k,c(a_k)} \leftarrow \mathbf{C}_{k-1,c(a_k)} + \mathbf{x}_{a_k}\mathbf{z}^T_{a_k}$
17: $\mathbf{B}_{k,c(a_k)} \leftarrow \mathbf{B}_{k-1,c(a_k)} + \mathbf{x}_{a_k}\widetilde{r}_k$
18: $\mathbf{P}_k \leftarrow \mathbf{P}_k + \mathbf{z}_{a_k}\mathbf{z}^T_{a_k} - \mathbf{C}^T_{k,c(a_k)}\mathbf{A}^{-1}_{k,c(a_k)}\mathbf{C}_{k,c(a_k)}$
19: $\mathbf{Q}_k \leftarrow \mathbf{Q}_k + \mathbf{z}_{\tau,a_k}\widetilde{r}_k - \mathbf{C}^T_{k,c(a_k)}\mathbf{A}^{-1}_{k,c(a_k)}\mathbf{B}_{k,c(a_k)}$
20: $\hat{\boldsymbol{\rho}}_{k+1} \leftarrow \mathbf{P}_k\mathbf{Q}_k$
21: $o_{c(a_k)} += 1$, $o_k += 1$, $z^{c(a_k)}_{a_k} += 1$
22: **for** $c \in \{1, \ldots, C\}$ **do**
23: $\hat{\boldsymbol{\theta}}_{k+1,c} \leftarrow \mathbf{A}^{-1}_{k,c}(\mathbf{B}_{k,c} - \mathbf{C}_{k,c}\hat{\boldsymbol{\rho}}_{k+1})$
24: $\hat{b}_{k+1,c} \leftarrow \bar{r}_{k+1,c} - \bar{f}(\mathbf{x}_a, \hat{\boldsymbol{\theta}}_{k+1,c}) - \bar{g}(\mathbf{z}_a, \hat{\boldsymbol{\rho}}_{k+1})$
25: Update α_k, β_k, γ_k based on Lemma 1, 2 and 3
26: $\mathcal{D}_t \leftarrow \mathcal{D}_t - \{a_k\}$
27: Detect anomalies by $\hat{\mathbf{r}}_{t+1}, a$ before the $(t+1)$-th update

B. TS-Based Strategy

Thompson Sampling [22] is another popular criteria to meet the balance between exploitation and exploration, which selects arms by sampling from the posterior distribution of the optimal arm on candidate arms. As the optimal arm is unknown, modeling the distribution of expected reward is adopted. In each time step, we sample a reward from the posterior distribution of expected reward for each arm, present the arm with the largest sampled reward to the experts, receive the label, and utilize the received labels to update the reward distribution. As the optimal arm receives the highest expected reward, such a sampling process is equivalent to direct sampling from the posterior distribution of the optimal arm on candidate arms.

For TS, Bayesian regret is widely used to minimize expectation of accumulated regret $Reg(TK)$ as

$$BayesRegret(TK) = \mathbb{E}(Reg(TK)). \quad (20)$$

Also, there are two main steps in the k-th interaction:

- *Selection Policy:* Sample reward $\hat{r}_{k,a}$ by Gaussian distribution based on the expected rewards, and find the arm a_k with highest $\hat{r}_{k,a_k}$ for query label from experts; and
- *Parameter Update:* Update parameters based on the received reward $\widetilde{r}_k$ for the Gaussian distribution in the next interaction.

The process is similar to the UCB-based methods, but the strategy to explore new possibilities is different. For UCB-based algorithms, the size of region for exploration depends on the Upper Confidence Bounds, while it depends on the variance of Gaussian distribution in this section. We will introduce the selection policy first.

1) Selection Policy: Based on Eq. (7), the expected reward $\hat{r}_{k,a}$ of each arm a in time k can be regarded as a sample from the Gaussian distribution $N(f(\mathbf{x}_a, \hat{\boldsymbol{\theta}}_{k,c(a)}) + g(\mathbf{z}_a, \hat{\boldsymbol{\rho}}_k), \sigma_1^2)$, where the bias $\hat{b}_{k,c(a)}$ is controlled by the variance σ_1.

In order to minimize the $BayesRegret(TK)$, we update our model by $P(\hat{\boldsymbol{\theta}}_{k,c}, \hat{\boldsymbol{\rho}}_k|\mathcal{F}_k)$, and update the expected reward estimation by $P(\hat{r}_{k,a}|\mathcal{F}_k, \hat{\boldsymbol{\theta}}_{k,c(a)}, \hat{\boldsymbol{\rho}}_k)$ with updated parameters. Based on Bayesian function, the posterior distribution of parameters $P(\hat{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\rho}}_k|\mathcal{F}_k)$ and the posterior distribution of the expected reward $P(\hat{r}_{k,a}|\mathcal{F}_k, \hat{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\rho}}_k)$ can be denoted as

$$\begin{aligned} P(\hat{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\rho}}_k|\mathcal{F}_k) \propto & \prod_{\tau=1}^{k-1} P(\widetilde{r}_\tau|\hat{r}_{\tau,a_\tau}) \\ & \prod_{a\in\mathcal{D}} P(\hat{r}_{\tau,a}|\hat{\boldsymbol{\theta}}_{\tau,c(a)}, \hat{\boldsymbol{\rho}}_\tau) \\ & \times P(\hat{\boldsymbol{\theta}}_{\tau,c(a)}, \hat{\boldsymbol{\rho}}_\tau), \end{aligned} \tag{21}$$

$$\begin{aligned} P(\hat{r}_{k,a}|\mathcal{F}_k, \hat{\boldsymbol{\theta}}_{k,c(a)}, \hat{\boldsymbol{\rho}}_k) \propto & \prod_{\tau\in\mathcal{C}_k} P(\widetilde{r}_\tau|\hat{r}_{\tau,a_\tau}) \\ & \times P(\hat{r}_{\tau,a}|\hat{\boldsymbol{\theta}}_{\tau,c(a)}, \hat{\boldsymbol{\rho}}_\tau). \end{aligned} \tag{22}$$

where $\mathcal{C}_k = \{\tau < t : c(a_\tau) = c(a)\}$.

2) Parameter Update: Except obtaining the linear expected reward $\hat{r}_{k,a}|\hat{\boldsymbol{\theta}}_{k,c(a)}, \hat{\boldsymbol{\rho}}_k$ from $N(\mathbf{x}_a^T\hat{\boldsymbol{\theta}}_{k,c(a)} + \mathbf{z}_a^T\hat{\boldsymbol{\rho}}_k, \sigma_1^2)$, the distribution of received reward $P(\widetilde{r}_k|\hat{r}_{k,a_k})$, and the distributions of parameters $P(\hat{\boldsymbol{\theta}}_{k,c})$ and $P(\hat{\boldsymbol{\rho}}_k)$ can be also be regarded as samples from Gaussian distributions based on [20], [37],

$$\begin{aligned} \widetilde{r}_k|\hat{r}_{k,a_k} &\sim N(\hat{r}_{k,a_k}, \sigma_2^2), \\ \hat{\boldsymbol{\theta}}_{k,c} &\sim N(\mathbf{0}, \sigma_3^2\mathbf{I}_d), \\ \hat{\boldsymbol{\rho}}_k &\sim N(\mathbf{0}, \sigma_4^2\mathbf{I}_C), \end{aligned} \tag{23}$$

where σ_1, σ_2, σ_3 and σ_4 are hyper-parameters. Inserting the Eq. (23) and Eq. (7) into Eq. (21), we can get the mean $\bar{\boldsymbol{\theta}}_{k,c}$ of the parameter posterior distribution by setting its probability density function's derivative to zero $\frac{\partial P(\hat{\boldsymbol{\theta}}_{k,c}, \hat{\boldsymbol{\rho}}_k|\mathcal{F}_k)}{\partial \hat{\boldsymbol{\theta}}_{k,c}} = 0$ as

$$\begin{aligned} \sum_{\tau\in\mathcal{C}_k} \frac{1}{\sigma_1^2}\mathbf{x}_{a_\tau}(-\bar{r}_{\tau,c(a_\tau)} + \mathbf{x}_{a_\tau}^T\hat{\boldsymbol{\theta}}_{\tau,c(a_\tau)} + \mathbf{z}_{a_\tau}^T\hat{\boldsymbol{\rho}}_\tau) \\ + \frac{1}{\sigma_3^2}\hat{\boldsymbol{\theta}}_{\tau,c(a_\tau)} = 0 \end{aligned} \tag{24}$$

Thus, we have $\bar{\boldsymbol{\theta}}_{k,c(a)} = \mathbf{D}_{k,c(a)}^{-1}\mathbf{E}_{k,c(a)}$, where

$$\begin{aligned} \mathbf{D}_{k,c(a)} &= \sum_{\tau\in\mathcal{C}_k} \frac{1}{\sigma_1^2}\mathbf{x}_{a_\tau}\mathbf{x}_{a_\tau}^T + \frac{1}{\sigma_3^2}\mathbf{I}_d, \\ \mathbf{E}_{k,c(a)} &= \sum_{\tau\in\mathcal{C}_k} \frac{(\bar{r}_{\tau,c(a_\tau)} - \mathbf{z}_{a_\tau}^T\hat{\boldsymbol{\rho}}_\tau)\mathbf{x}_{a_\tau}}{\sigma_1^2}. \end{aligned} \tag{25}$$

The deviation can be obtained by calculating the second derivative of $P(\hat{\boldsymbol{\theta}}_{k,c}, \hat{\boldsymbol{\rho}}_k|\mathcal{F}_k)$ to $\hat{\boldsymbol{\theta}}_{k,c}$, i.e., $\mathbf{D}_{k,c}^{-1}$, and we have

$$\hat{\boldsymbol{\theta}}_{k,c}|\mathcal{F}_k \sim N(\bar{\boldsymbol{\theta}}_{k,c}, \mathbf{D}_{k,c}^{-1}). \tag{26}$$

Similarly, we have

$$\hat{\boldsymbol{\rho}}_k|\mathcal{F}_k \sim N(\bar{\boldsymbol{\rho}}_k, \mathbf{F}_k^{-1}), \tag{27}$$

Algorithm 3 TS_HADDN

Input: $\mathbf{X}$, $\mathcal{P}$, $\mathcal{D}_t$, K, C, d, δ_1, δ_2, δ_3, δ_4

// Initialization
for $c \in C$ **do**
  Sample $\boldsymbol{\theta}_{1,c}$ from Eq. (23)
Sample $\boldsymbol{\rho}_1$ from Eq. (23)
// In the t-th communication network
for k = 1, …, K **do**
  // Select the presented flow based on $\hat{r}_{t,a}$
  for $a \in \mathcal{D}_t$ **do**
    Sample $\hat{r}_{k,a}$ from $N(\bar{r}_{k,a}, \sigma_{k,a}^2)$
  Choose $a_k = \arg\max_{a\in\mathcal{D}} \hat{r}_{k,a}$ and get the feedback $\widetilde{r}_k$
  // Parameter update
  $\mathbf{D}_{k,c(a_k)} \leftarrow \mathbf{D}_{k-1,c(a_k)} + \frac{1}{\sigma_1^2}\mathbf{x}_{a_k}\mathbf{x}_{a_k}^T$
  $\mathbf{E}_{k,c(a_k)} \leftarrow \mathbf{E}_{k-1,c(a_k)} + \frac{(\widetilde{r}_k - \mathbf{z}_{a_k}^T\hat{\boldsymbol{\rho}}_{k-1})\mathbf{x}_{a_k}}{\sigma_1^2}$
  Sample $\hat{\boldsymbol{\theta}}_{k,c(a_k)}$ from $N(\mathbf{D}_{k,c(a_k)}^{-1}\mathbf{E}_{k,c(a_k)}, \mathbf{D}_{k,c(a_k)}^{-1})$
  for $c \in C - \{c(a_k)\}$ **do**
    $\mathbf{D}_{k,c} \leftarrow \mathbf{D}_{k-1,c}$, $\mathbf{E}_{k,c} \leftarrow \mathbf{E}_{k-1,c}$
  $\mathbf{F}_k \leftarrow \mathbf{F}_{k-1} + \frac{1}{\sigma_1^2}\mathbf{z}_{a_k}\mathbf{z}_{a_k}^T$
  $\mathbf{G}_k \leftarrow \mathbf{G}_{k-1} + \frac{(\widetilde{r}_k - \mathbf{x}_{a_k}^T\hat{\boldsymbol{\theta}}_{k-1,c(a_k)})}{\sigma_1^2}\mathbf{z}_{a_k}$
  Sample $\hat{\boldsymbol{\rho}}_k$ from $N(\mathbf{F}_k^{-1}\mathbf{G}_k, \mathbf{F}_k^{-1})$
  $o_{c(a_k)} \leftarrow o_{c(a_k)} + 1$
  $\mathcal{D}_t \leftarrow \mathcal{D}_t - \{a_k\}$
Detect anomalies by $\hat{\mathbf{r}}_{t+1}, a$ before the $(t+1)$-th update

where

$$\begin{aligned} \bar{\boldsymbol{\rho}}_k &= \mathbf{F}_k^{-1}\mathbf{G}_k, \\ \mathbf{F}_k &= \sum_{\tau}^{k-1} \frac{1}{\sigma_1^2}\mathbf{z}_{a_\tau}\mathbf{z}_{a_\tau}^T + \frac{1}{\sigma_4^2}\mathbf{I}_C, \\ \mathbf{G}_k &= \sum_{\tau}^{k-1} \frac{(\bar{r}_{\tau,c(a_\tau)} - \mathbf{x}_{a_\tau}^T\hat{\boldsymbol{\theta}}_{\tau,c(a_\tau)})}{\sigma_1^2}\mathbf{z}_{a_\tau}. \end{aligned} \tag{28}$$

Next, we insert Eq. (23) into Eq. (22), and the mean $\bar{r}_{k,a}$ of expected reward posterior distribution by calculating $\frac{\partial P(\hat{r}_{k,a}|\mathcal{F}_k, \hat{\boldsymbol{\theta}}_{k,c(a)}, \hat{\boldsymbol{\rho}}_k)}{\partial \hat{r}_{k,a}} = 0$ as

$$-o_{c(a)}\sigma_1^2(\bar{r}_{k,a} - \bar{r}_{k,c(a)}) - \sigma_2^2(\bar{r}_{k,a} - \mathbf{x}_a^T\hat{\boldsymbol{\theta}}_{k,c(a)} - \mathbf{z}_a^T\hat{\boldsymbol{\rho}}_k) = 0 \tag{29}$$

Thus, we have

$$\bar{r}_{k,a} = \frac{o_{c(a)}\sigma_1^2\bar{r}_{k,c(a)} + \sigma_2^2(\mathbf{x}_a^T\hat{\boldsymbol{\theta}}_{k,c(a)} + \mathbf{z}_a^T\hat{\boldsymbol{\rho}}_k)}{o_{c(a)}\sigma_1^2 + \sigma_2^2} \tag{30}$$

Also, based on the second derivative of $P(\hat{r}_{k,a}|\mathcal{F}_k, \hat{\boldsymbol{\theta}}_{k,c(a)}, \hat{\boldsymbol{\rho}}_k)$ to $\hat{r}_{k,a}$, the deviation is

$$\sigma_{k,a}^2 = (\frac{1}{\sigma_1^2} + \frac{o_{c(a)}}{\sigma_2^2})^{-1}. \tag{31}$$

Thus, we have

$$\hat{r}_{k,a}|\mathcal{F}_k, \hat{\boldsymbol{\theta}}_{k,c(a)}, \hat{\boldsymbol{\rho}}_k \sim N(\bar{r}_{k,a}, \sigma_{k,a}^2). \tag{32}$$

3) TS-Based Online Learning Algorithm: With the TS-based strategy, we *exploit* historical feedback to calculate the mean expected reward in Eq. (30) and *explore* new possibility by the variance of Gaussian distribution in Eq. (32). We summarize the TS-based online learning in Algorithm 3. In each interaction, the expected reward of each flow is sampled in line 9 by Eqs. (31) and (32). After we present the arm with the highest $\hat{r}_{k,a_k}$, we get feedback from experts and update parameters from line 12-19 based on Eqs. (25) and (28).

Based on Theorem 3, we bound the regret of TS_HADDN to $\tilde{O}(\sqrt{TKd})$. Similar to UCB_HADDN, although it is a common upper bound of regret, we can achieve better anomaly detection performance with well designed expected reward and periodic labels in the dynamic communication networks.

Theorem 3: With probability at least $\frac{4\delta}{NT^2K^2}$*, the upper Bayesian regret bound of the TS_HADDN is*

$$\tilde{O}(\sqrt{NTK}(\frac{d}{\sqrt{N}})^{\alpha}) \tag{33}$$

Proof: Based on Proposition 1 of [44], the bayesian regret can be obtained by

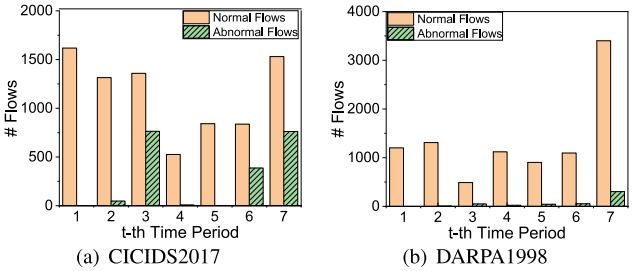
$$\begin{aligned} BayesRegret(TK) &= \mathbb{E}(\mathbf{U}_{k,a_k} - \hat{r}_{k,a_k}) + \mathbb{E}(r^*_k - \mathbf{U}_{k,a_k}) \\ &\leq O(\sum_{t=1}^{T}\sum_{k=1}^{K}|\hat{r}_{k,a_k} - r^*_k|) \end{aligned} \tag{34}$$

Based on Eq. (30), we have

$$\begin{aligned} |\bar{r}_{k,a_k} - r^*_k| &\leq \frac{o_{c(a_k)}\sigma_1^2\bar{r}_{k,c(a_k)} + \sigma_2^2(\mathbf{x}_{a_k}^T\hat{\boldsymbol{\theta}}_{k,c(a_k)} + \mathbf{z}_{a_k}^T\hat{\boldsymbol{\rho}}_k)}{o_{c(a_k)}\sigma_1^2 + \sigma_2^2} \\ &\quad + (\frac{o_{c(a_k)}\sigma_1^2}{o_{c(a_k)}\sigma_1^2 + \sigma_2^2} + \frac{\sigma_2^2}{o_{c(a_k)}\sigma_1^2 + \sigma_2^2})r^*_k \\ &\leq \frac{o_{c(a_k)}\sigma_1^2}{o_{c(a_k)}\sigma_1^2 + \sigma_2^2}|\bar{r}_{k,c(a_k)} - r^*_k| \\ &\quad + \frac{\sigma_2^2}{o_{c(a_k)}\sigma_1^2 + \sigma_2^2}|\mathbf{x}_{a_k}^T\hat{\boldsymbol{\theta}}_{k,c(a_k)} - \mathbf{x}_{a_k}^T\boldsymbol{\theta}^*_{c(a_k)}| \\ &\quad + \frac{\sigma_2^2}{o_{c(a_k)}\sigma_1^2 + \sigma_2^2}|\mathbf{z}_{a_k}^T\hat{\boldsymbol{\rho}}_k - \mathbf{z}_{a_k}^T\boldsymbol{\rho}^*| \\ &\quad + \frac{\sigma_2^2}{o_{c(a_k)}\sigma_1^2 + \sigma_2^2}|\mathbf{x}_{a_k}^T\hat{\boldsymbol{\theta}}_{k,c(a_k)} + \mathbf{z}_{a_k}^T\hat{\boldsymbol{\rho}}_k - r^*_k|. \end{aligned} \tag{35}$$

Based on Lemma 5 of [20], with probability $1 - \frac{4\delta}{NT^2K^2}$, it is easy to obtain that

$$\begin{aligned} &|\hat{r}_{k,a_k} - r^*_k| \leq |\hat{r}_{k,a_k} - \bar{r}_{k,c(a_k)}| + |\bar{r}_{k,c(a_k)} - r^*_k| \\ &\leq \sqrt{2\ln\frac{NT^2K^2}{2\delta}}\sigma_{k,a_k} + \frac{o_{c(a_k)}\sigma_1^2}{o_{c(a_k)}\sigma_1^2 + \sigma_2^2}R \\ &\quad \times\sqrt{\frac{2\ln 2NT^2K^2/\delta}{o_{c(a_k)}}} \\ &\quad + \frac{\sigma_2^2}{o_{c(a_k)}\sigma_1^2 + \sigma_2^2}((||\mathbf{x}_{a_k}||_{\mathbf{D}^{-1}_{k,c(a_k)}} + ||\mathbf{z}_{a_k}||_{\mathbf{F}_k^{-1}}) \\ &\quad \times(\sqrt{\frac{NTK}{N\sigma_2^2 + TK\sigma_1^2}}d_{max} \\ &\quad + 2R\sqrt{\frac{2d\ln NT^2K^2}{2\delta}}) + d_{max}). \end{aligned} \tag{36}$$



(a) CICIDS2017 (b) DARPA1998

Fig. 4. Distribution of normal/abnormal flows over different time periods.

where we define $d_{max} \leq \sqrt{N/dTK}(\frac{d}{\sqrt{N}})^{2\alpha}$ and $\frac{\sigma_2^2}{\sigma_1^2} = \frac{TK}{N}(\frac{d}{\sqrt{N}})^{-\alpha}$, and we can get the upper bound of $BayesRegret(TK)$ as

$$\begin{aligned} &O(\sum_{t=1}^{T}\sum_{k=1}^{K}|\hat{r}_{k,a_k} - r^*_k|) \\ &\leq \tilde{O}(\frac{\sigma_2^2}{o_{c(a_k)}\sigma_1^2 + \sigma_2^2}(||\mathbf{x}_{a_k}||_{\mathbf{D}^{-1}_{k,c(a_k)}} + ||\mathbf{z}_{a_k}||_{\mathbf{F}_k^{-1}}) \\ &\quad \times\sqrt{\frac{NTK}{N\sigma_2^2 + TK\sigma_1^2}}d_{max}) \\ &\leq \tilde{O}(\frac{\sigma_2^2}{\sigma_1^2}N\ln(\sigma_2^2 + \frac{TK}{N}\sigma_1^2)\sqrt{d}d_{max}) \\ &\leq \tilde{O}(\sqrt{NTK}(\frac{d}{\sqrt{N}})^{\alpha}) \end{aligned} \tag{37}$$

which completes the proof. □

VII. Experimental Analysis

In this section, we conduct experiments to demonstrate the effectiveness of our proposed framework HADDN. Through the experiments, we aim to answer the following questions:

- *Q1:* Compared to the state-of-the-art models, can the proposed UCB_HADDN and TS_HADDN achieve better labeling performance in the dynamic networks?
- *Q2:* Can UCB_HADDN and TS_HADDN achieve better anomaly detection performance?
- *Q3:* How do the abnormal and normal labels affect the final anomaly detection performance? and
- *Q4:* How do the pre-defined parameters affect the anomaly detection performance?

Next, we will first introduce the experiment settings followed by experiments to answer these questions.

A. Experimental Settings

1) Datasets: Two publicly available datasets CICIDS2017[1] and DARPA1998[2] are used for evaluation. Following the guidance,[3] we first extract 41 features for data packages of each dataset, including continuous features like flow duration and categorical features like protocols. For those categorical ones, we use one-hot vectors to represent these features.

[1]https://www.unb.ca/cic/datasets/ids-2017.html

[2]https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset

[3]http://kdd.ics.uci.edu/databases/kddcup99/

TABLE II
STATISTICS OF DATASETS

	CICIDS2017		DARPA1998	
	$\mathcal{D}_t$	$\mathcal{D}_{t+1}$	$\mathcal{D}_t$	$\mathcal{D}_{t+1}$
# Normal Flows	6497	1532	6166	3401
# Abnormal Flows	1207	762	177	304

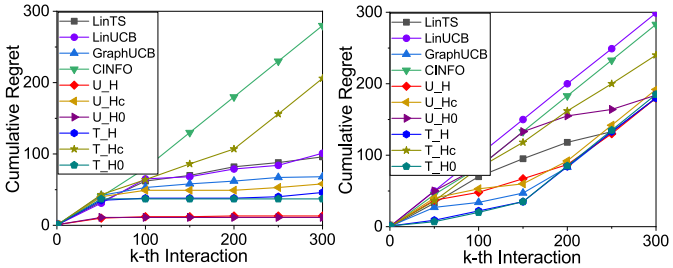
To imitate the dynamic network environment, one day and one week are regarded as a new time period for CICIDS2017 and DARPA1998, respectively, thus both datasets have seven time periods. Also, to provide enough interactions for algorithm convergence as well as evaluate the adaptiveness in the new time period, we divide the datasets into two parts $\mathcal{D}_t$ and $\mathcal{D}_{t+1}$, where $\mathcal{D}_t$ is made up of flows before the last time period and $\mathcal{D}_{t+1}$ is constructed by flows of the last time period. What's more, to estimate whether we can explore new attacks in the new time period, we imitate new attacks. On the $\mathcal{D}_{t+1}$, the most frequent attacks are "PortScan" and "satan" for CICIDS2017 and DARPA1998, respectively. Thus we regard "PortScan" and "satan" as new attacks, and remove all flows of these two attack types on $\mathcal{D}_t$ for both datasets. Also, to estimate whether we can detect old attacks based on the historical labels, we randomly select about 10% normal and abnormal flows from each time period into $\mathcal{D}_{t+1}$.

After the preprocessing, the distributions of abnormal/normal flows over different time periods are shown in Fig. 4 and the detailed statistics are shown in Table II. From which, we can observe that attacks do not always occur as bursts (e.g., 2nd period in CICIDS2017 and 4th period in DARPA1998), and abnormal flows are much smaller than normal flows generally.

2) Compared Methods: We compare with the representative and state-of-the-art contextual approaches, which include

- *LinUCB [36]:* A contextual bandit algorithm whose linear parametric expected rewards utilize flow features to learn separate parameters for each arm.
- *LinTS [37]:* A linear parametric Thompson-sampling-based bandit algorithm whose parameters are globally shared among all flows.
- *GraphUCB [11]:* A LinUCB-based anomaly detection model, where their parameters can be shared among instances in the same feature-based clusters.
- *CINFO [4]:* An unsupervised anomaly detection model with feature selection.
- *Meta-AAD [12]:* A reinforcement learning model which can adaptively detect anomalies from different distributions of data.
- *U_Hc:* Our UCB_HADDN but eliminating the structure correlation parts in the expected reward and upper bound.
- *U_H0:* Our UCB_HADDN without the exploration part.
- *T_Hc:* Our TS_HADDN but eliminating the structure correlation parts in the expected reward and Gaussian distribution.
- *T_H0:* Our TS_HADDN without the exploration part.

For simplification, we also use U_H and T_H to represent our proposed UCB_HADDN and TS_HADDN in the following experiments. Since GraphUCB is developed for attributed networks, it assumes that linked instances share similarities, which doesn't hold for the communication networks. Thus we only implement the nodal attributes part of their algorithm, i.e., Eq. (4) in our paper. This approach can also be treated as a variant of UCB_HADDN, which utilizes the cluster-based parameters but eliminates the non-parametric deduction and structural correlations. Also, for the unsupervised approach CINFO, the interactive process can be regarded as that we present abnormal flows to the experts by their abnormality estimation results in descending order. As to Meta-AAD, it requires batches of labels to train the adaptive model, and can not be used for labeling. Thus, we directly provide the maximized number of true labels to train the model and only compare with it in Section VII-C. Also, in order not to influence the new abnormal patterns of $\mathcal{D}_{t+1}$ and provide fair results, we update the randomly selected 10% part of $\mathcal{D}_{t+1}$ for five times and present the average results.



(a) CICIDS2017 (b) DARPA1998

Fig. 5. Labeling performance comparison on $\mathcal{D}_t$.

3) Evaluation Metrics: To evaluate both the labeling performance and the anomaly detection performance, we adopt several widely used metrics.

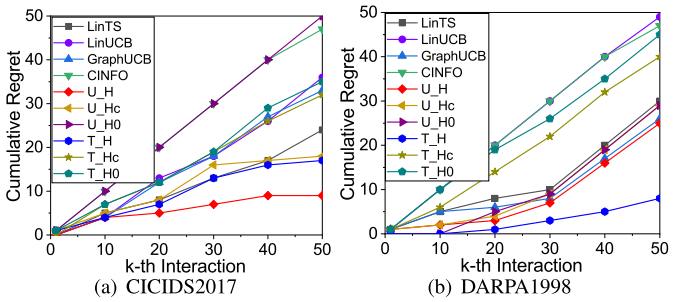
- *Cumulative Regret [19], [24]:* To evaluate the labeling performance on both $\mathcal{D}_t$ and $\mathcal{D}_{t+1}$, this metric shows the cumulative regret we receive with respect to the number of interactions by Eq.(3). Lower cumulative regret represents higher labeling performance.
- *Anomaly Discovery Curve [11], [33]:* To evaluate the anomaly detection performance on the $\mathcal{D}_{t+1}$, it plots the number of true anomalies with respect to n. Each dot in this curve represents the number of true anomalies among those with the top n abnormality estimation values. Ideally, this curve should climb as quickly as possible.

B. Labeling Performance Comparison

To answer the **Q1**, we will evaluate the labeling performance on both $\mathcal{D}_t$ and $\mathcal{D}_{t+1}$. In order to eliminate the noisy features for building clusters, we use MCFS [40] to select features and utilize KMeans to construct clusters. For CICIDS2017, $d = 10$ and $C = 10$; and for DARPA1998, $d = 20$ and $C = 5$.

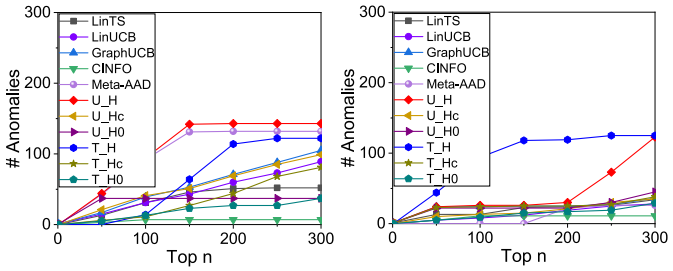
We provide 50 labels for each time period. As we have six time periods in $\mathcal{D}_t$, we provide 300 labels on $\mathcal{D}_{t+1}$ in both datasets, and k changes from 0 to 300. Similarly, there are 50 labels on $\mathcal{D}_{t+1}$ and k changes from 0 to 50 on $\mathcal{D}_{t+1}$. The comparison results on cumulative regret in two datasets are presented in Fig. 5 and Fig. 6. From these results, we make the following observations

- Generally, the interactive learning algorithm performs better than the unsupervised CINFO, which demonstrates the necessity of experts' involvement.
- LinUCB sometimes plays worse than CINFO, which is because it trains each arm separately and suffers from cold start. It also demonstrates the necessity of clustering.
- Generally, our proposed approaches can outperform other state-of-the-art algorithms in both datasets especially on



(a) CICIDS2017 (b) DARPA1998

Fig. 6. Labeling performance comparison on $\mathcal{D}_{t+1}$.



(a) CICIDS2017 (b) DARPA1998

Fig. 7. Anomaly detection performance comparison on $\mathcal{D}_{t+1}$.

$\mathcal{D}_{t+1}$ in Fig. 6, which shows that we can gain better tradeoff between explore and exploitation and achieve better labeling performance on $\mathcal{D}_t$.

- UCB_HADDN provides more labels for anomalies than U_Hc in general, and so are TS_HADDN and T_Hc, which demonstrates the effectiveness of utilizing structural correlations.
- In CICIDS2017, U_H0 and U_H receive similar cumulative regrets. In DARPA1998, T_H0 and T_H achieve matchable performance. This is because some attacks can lead to great numbers of abnormal flows. Without exploration, H_H0 and T_H0 can fully exploit historical labels and provide right labels for flows in the same abnormal patterns. However, U_H0 and T_H0 can not quickly adapt to new abnormal patterns on $\mathcal{D}_{t+1}$ (as shown in Fig. 6 and Fig. 7).

Based on the aforementioned observations of both $\mathcal{D}_t$ and $\mathcal{D}_{t+1}$, we can answer the **Q1** that our UCB_HADDN and TS_HADDN can achieve better labeling performance in the dynamic networks.

C. Anomaly Detection Performance Comparison

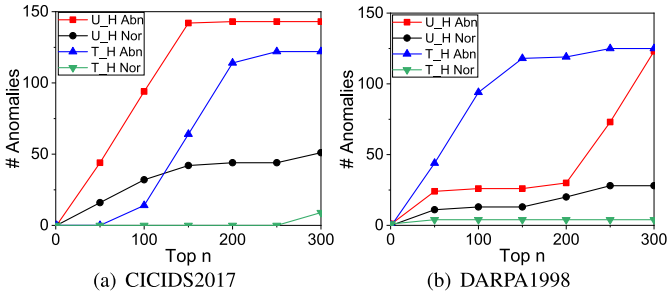
To answer the **Q2**, we utilize the model learned from $\mathcal{D}_t$, and estimate the abnormality of each flow on $\mathcal{D}_{t+1}$ based on the expected reward (i.e., abnormality estimation) of each algorithm. In Fig. 7, we plot the anomaly discovery curve to show how the number of anomalies changes with regard to n, where each dot represents the number of true anomalies among those with the top n abnormality estimation results. Also, in Table III, we provide the time (seconds) we use to provide 300 labels on $\mathcal{D}_t$ (i.e., train the model) and the time to obtain the abnormality estimation results on $\mathcal{D}_{t+1}$ (i.e., test the detection performance). From these results, we can observe that

- As the unsupervised CINFO only treats the $\mathcal{D}_{t+1}$ as a new static network, the interactive learning algorithm can detect more abnormal flows with the updated model based on experts' historical guidance.
- Among the interactive learning algorithms, our U_H and T_H consistently outperform other baselines, demonstrating the effectiveness of involving the cluster-based non-parametric value.
- Among all baselines, Meta-AAD can obtain similar results to our U_H in CICIDS2017. However, as shown in Table III, with complex neural networks, Meta-AAD needs much more time to train the model. Also, as Meta-AAD requires large numbers of labels for training, the performance in DARPA1998 is not as good as ours.

Based on the aforementioned observations, we can answer the **Q2** that our UCB_HADDN and TS_HADDN can achieve better anomaly detection performance compared to the state-of-the-art models.

TABLE III
EFFICIENCY COMPARISON

Time (seconds)	CICIDS2017		DARPA1998	
	Train	Test	Train	Test
LinTS	4.39	0.02	3.95	0.02
LinUCB	13.80	0.01	11.01	0.01
GraphUCB	14.26	0.02	12.01	0.02
CINFO	2.23	1.54	1.56	1.53
Meta-AAD	151.53	3.54	174.68	5.89
U_H	33.95	0.05	23.99	0.03
T_H	56.64	0.06	41.70	0.09



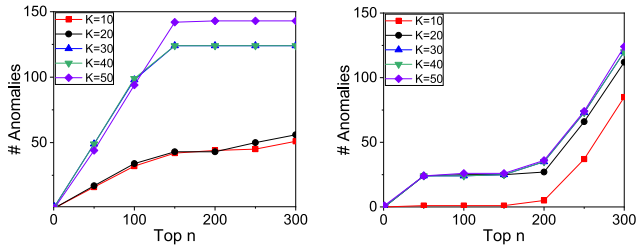
(a) CICIDS2017 (b) DARPA1998

Fig. 8. Anomaly detection performance with different strategies.

D. Influence of the Labeled Normal and Abnormal

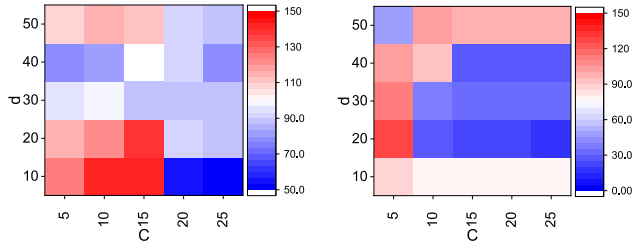
To provide insights of the importance of labeled normal and abnormal flows, we change the selection policy to maximize the number of labeled normal flows on $\mathcal{D}_t$, i.e., if we present a normal flow to the expert, we receive a reward 1, otherwise the reward is 0. After the labeling process with changed selection policy on $\mathcal{D}_t$, we estimate the abnormality on $\mathcal{D}_{t+1}$ with the updated model. The comparison results are presented in Fig. 8. We can observe better anomaly detection performance when we try to maximize the labeled anomalies on $\mathcal{D}_t$.

With the original strategy to maximize the number of labeled anomalies on $\mathcal{D}_t$, we further estimate how the maximized number of labels K in each time period influences the anomaly detection performance on $\mathcal{D}_{t+1}$. As shown in Fig. 9, more labels can provide better anomaly detection performance in general. However, as some attacks can lead to large numbers of abnormal flows and there is tradeoff between exploitation



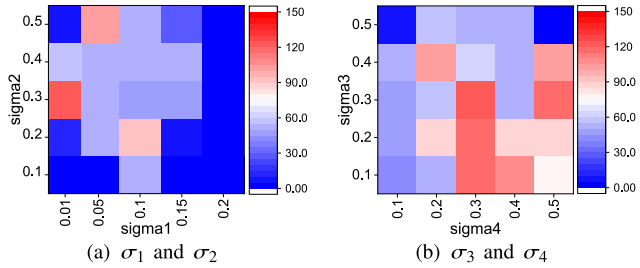
(a) CICIDS2017 (b) DARPA1998

Fig. 9. Anomaly detection performance with different numbers of labels.



(a) UCB_HADDN (b) TS_HADDN

Fig. 10. Impacts of C and d on anomaly detection performance.



(a) σ_1 and σ_2 (b) σ_3 and σ_4

Fig. 11. Impacts of σ_1, σ_2, σ_3 and σ_4 on anomaly detection performance.

and exploration, the increase of labels sometimes can not lead to an increase in anomaly detection performance.

Thus, we can answer the **Q3**, despite the tradeoff between exploitation and exploration weaken the effectiveness of labels, compared to the labeled normal flows, the labeled anomalies can provide more help to adapt to dynamic networks and obtain better anomaly detection performance, which also provides evidence for our initial target to maximize the labeled abnormal flows in dynamic networks.

E. Parameter Analysis

With the same proposed expected reward, both implementation approaches have two pre-defined parameters, i.e., the number of clusters C and feature dimension d. Specially, for TS_HADDN, there are four extra parameters, i.e., σ_1, σ_2, σ_3 and σ_4, to denote the variance of expected reward $\hat{r}_{k,a}$, the received reward $\widetilde{r}_k$, the feature weight $\hat{\theta}_{k,c}$ and the structure weight $\hat{\rho}_k$.

In Fig. 10 and Fig. 11, we only show the results in CICIDS2017 since we have similar observations on both datasets. For C and d in Fig. 10, although the mutual influence is complicated and not monotonic, with the increase of C, the performance generally first increases, implying theeffectiveness of clustering. For the four variances in Fig. 11, as σ_1 and σ_2 are related to reward and σ_3 and σ_4 are related to parameters, we divide them into two groups for simplification. We can observe that with the increase of the four variances, the performance generally first increases, suggesting wider exploration on $\mathcal{D}_t$ can help improve the anomaly detection performance on $\mathcal{D}_{t+1}$. After the first increase, the performance decreases, implying that too wide exploration may lead to far deviated from the exploitation results (i.e., the mean values). There is a tradeoff between exploration and exploitation.

VIII. Conclusion

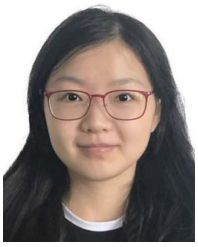
In this paper, we novelly utilize semi-parametric bandits to detect abnormal flows in dynamic networks with limited labor resources. We propose a novel semi-parametric bandit framework HADDN, utilizing contextual information, i.e., feature-based clusters and structural correlations, to adapt to dynamic networks, making connections between historical labels and new emerging flows. The proposed semi-parametric bandit framework leverages parametric functions with contextual information to ensure the efficiency of anomaly detection, and takes advantage of non-parametric value to improve the accuracy of anomaly detection by the closed gap between the expected reward and real reward. We provide two linear implementations UCB_HADDN and TS_HADDN for HADDN with theoretical proof. Experimental results on two publicly available datasets demonstrate the great improvement of HADDN compared to other state-of-the-art baselines.

There are several interesting directions that need further investigation as future work. First, in this paper, we only present one flow in each interaction. We would like to present more flows in each interaction with lower computing complexity in streaming settings. Second, the number of clusters is fixed in this paper. We will study how to detect anomalies with flexible and scalable clustering strategies in dynamic networks with semi-parametric bandits next.

References

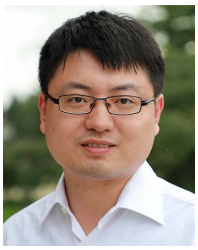
[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15:1–15:58, 2009.
[2] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, Jan. 2016.
[3] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: http://arxiv.org/abs/1901.03407
[4] G. Pang, L. Cao, L. Chen, D. Lian, and H. Liu, "Sparse modeling-based sequential ensemble learning for effective outlier detection in high-dimensional numeric data," in *Proc. AAAI*, 2018, pp. 3892–3899.
[5] D. Eswaran and C. Faloutsos, "SedanSpot: Detecting anomalies in edge streams," in *Proc. ICDM*, 2018, pp. 953–958.
[6] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, and D. Zhang, "Graph based tensor recovery for accurate internet anomaly detection," in *Proc. INFOCOM*, 2018, pp. 1502–1510.
[7] Y. Zhou, M. Han, L. Liu, J. S. He, and Y. Wang, "Deep learning approach for cyberattack detection," in *Proc. INFOCOM Workshops*, 2018, pp. 262–267.
[8] S. Mukkamala and A. H. Sung, "Detecting denial of service attacks using support vector machines," in *Proc. 12th IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, May 2003, pp. 1231–1236.
[9] M. Kloft and P. Laskov, "Online anomaly detection under adversarial impact," in *Proc. AISTATS*, 2010, pp. 405–412.
[10] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Inf. Sci.*, vol. 239, pp. 201–225, Aug. 2013.
[11] K. Ding, J. Li, and H. Liu, "Interactive anomaly detection on attributed networks," in *Proc. WSDM*, 2019, pp. 357–365.
[12] D. Zha, K. Lai, M. Wan, and X. Hu, "Meta-AAD: Active anomaly detection with deep reinforcement learning," in *Proc. ICDM*, 2020, pp. 771–780.

[13] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, "Anomaly detection in dynamic networks: A survey," *Wiley Interdiscipl. Rev., Comput. Stat.*, vol. 7, no. 3, pp. 223–247, 2015.
[14] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-based anomaly detection," 2018, *arXiv:1802.06222*. [Online]. Available: http://arxiv.org/abs/1802.06222
[15] G. Cormode and M. Thottan, *Algorithms for Next Generation Networks* (Computer Communications and Networks). Springer, 2010.
[16] X. Meng, S. Wang, Z. Liang, D. Yao, J. Zhou, and Y. Zhang, "Semi-supervised anomaly detection in dynamic communication networks," *Inf. Sci.*, vol. 571, pp. 527–542, Sep. 2021.
[17] J. Cannady, "Next generation intrusion detection: Autonomous reinforcement learning of network attacks," in *Proc. 23rd Nat. Inf. Syst. Secur. Conf.*, 2000, pp. 1–12.
[18] S. Shamshirband, B. Daghighi, N. B. Anuar, M. L. M. Kiah, A. Patel, and A. Abraham, "Co-FQL: Anomaly detection using cooperative fuzzy Q-learning in network," *J. Intell. Fuzzy Syst.*, vol. 28, no. 3, pp. 1345–1357, 2015.
[19] S. Li, B. Wang, S. Zhang, and W. Chen, "Contextual combinatorial cascading bandits," in *Proc. ICML*, 2016, pp. 1245–1253.
[20] M. Ou, N. Li, C. Yang, S. Zhu, and R. Jin, "Semi-parametric sampling for stochastic bandits with many arms," in *Proc. AAAI*, 2019, pp. 7933–7940.
[21] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2, pp. 235–256, 2002.
[22] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, pp. 285–294, Dec. 1933.
[23] A. N. Elmachtoub, R. McNellis, S. Oh, and M. Petrik, "A practical method for solving contextual bandit problems using decision trees," 2017, *arXiv:1706.0487*. [Online]. Available: https://arxiv.org/abs/1706.0487
[24] A. Krishnamurthy, Z. S. Wu, and V. Syrgkanis, "Semiparametric contextual bandits," in *Proc. ICML*, 2018, pp. 2781–2790.
[25] A. Ghosh, S. R. Chowdhury, and A. Gopalan, "Misspecified linear bandits," in *Proc. AAAI*, 2017, pp. 3761–3767.
[26] C. Gentile, S. Li, and G. Zappella, "Online clustering of bandits," in *Proc. ICML*, 2014, pp. 757–765.
[27] S. Li, A. Karatzoglou, and C. Gentile, "Collaborative filtering bandits," in *Proc. SIGIR*, 2016, pp. 539–548.
[28] X. Wang, S. C. H. Hoi, C. Liu, and M. Ester, "Interactive social recommendation," in *Proc. CIKM*, 2017, pp. 357–366.
[29] Y. Ban and J. He, "Local clustering in contextual multi-armed bandits," in *Proc. WWW*, 2021, pp. 2335–2346.
[30] H. Zhuang, C. Wang, and Y. Wang, "Identifying outlier arms in multi-armed bandit," in *Proc. NIPS*, 2017, pp. 5204–5213.
[31] Y. Ban and J. He, "Generic outlier detection in multi-armed bandit," in *Proc. KDD*, 2020, pp. 913–923.
[32] M. A. Siddiqui, A. Fern, T. G. Dietterich, R. Wright, A. Theriault, and D. W. Archer, "Feedback-guided anomaly discovery via online optimization," in *Proc. KDD*, 2018, pp. 2200–2209.
[33] S. Das, W. Wong, T. G. Dietterich, A. Fern, and A. Emmott, "Incorporating expert feedback into active anomaly discovery," in *Proc. ICDM*, 2016, pp. 853–858.
[34] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Inf. Sci.*, vol. 378, pp. 484–497, Feb. 2017.
[35] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, "Contextual bandits with linear payoff functions," in *Proc. AISTATS*, 2011, pp. 208–214.
[36] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. WWW*, 2010, pp. 661–670.
[37] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *Proc. ICML*, 2013, pp. 127–135.
[38] S. Alelyani, J. Tang, and H. Liu, "Feature selection for clustering: A review," in *Data Clustering*. London, U.K.: Chapman & Hall, 2018, pp. 29–60.
[39] Y. Peng *et al.*, "A practical semi-parametric contextual bandit," in *Proc. IJCAI*, 2019, pp. 3246–3252.
[40] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. KDD*, 2010, pp. 333–342.
[41] Y. Zhang, X. Luo, and H. Luo, "A multi-step attack-correlation method with privacy protection," *J. Commun. Inf. Netw.*, vol. 1, no. 4, pp. 133–142, Dec. 2016.
[42] E. Lughofer, "Extensions of vector quantization for incremental clustering," *Pattern Recognit.*, vol. 41, no. 3, pp. 995–1011, Mar. 2008.
[43] M. Ester, H. Kriegel, J. Sander, M. Wimmer, and X. Xu, "Incremental clustering for mining in a data warehousing environment," in *Proc. VLDB*, pp. 323–333.
[44] D. Russo and B. Van Roy, "Learning to optimize via posterior sampling," *Math. Oper. Res.*, vol. 39, no. 4, pp. 1221–1243, Nov. 2014.



Xuying Meng received the B.S. degree from Wuhan University in 2013 and the Ph.D. degree from the University of Chinese Academy of Sciences in 2018. She is currently an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences. She has published innovative works in top conference proceedings. Her current research interests include data mining and security protection of network services. She serves for numerous conference program committees.

Yequan Wang received the B.S. degree Tianjin University, China, in 2014, and the Ph.D. degree from Tsinghua University in 2019. He is currently an Assistant Professor with the Institute of Computing Technology, Chinese Academy of Sciences. He has published innovative works in top conference proceedings and journals. His research interests include sentiment analysis and data mining. He serves as a reviewer for top journals and conference program committees.



Suhang Wang (Member, IEEE) received the B.S. degree in electronics and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012, the M.S. degree in electrical engineering: systems from the University of Michigan, Ann Arbor, MI, USA, in 2013, and the Ph.D. degree in computer science from Arizona State University in 2018. He is currently an Assistant Professor with the College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA, USA. He has published innovative works in top conference proceedings and journals, such as WWW, AAAI, IJCAI, CIKM, SDM, WSDM, ICDM, CVPR, and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE). His research interests include graph mining, data mining, and machine learning. He serves for journal editorial boards and numerous conference program committees.

Di Yao received the B.S. degree from Northeastern University, China, in 2013, and the Ph.D. degree from the University of Chinese Academy of Sciences in 2019. He is currently an Assistant Professor with the Institute of Computing Technology, Chinese Academy of Sciences. He has published innovative works in top conference proceedings and journals. His research interests include spatio-temporal data mining and deep learning. He serves as a reviewer for top journals.

Yujun Zhang received the B.S. degree in computer science from Nankai University in 1999 and the Ph.D. degree in computer architecture from the University of Chinese Academy of Sciences in 2004. He is currently a Professor with the Institute of Computing Technology, Chinese Academy of Sciences, and the University of Chinese Academy of Sciences. His research interests include intelligent networking and systems, network architecture, and network measurement and testing. He received the Technological Invention Award from the China Computer Federation in 2013 and the Beijing Young Famous Teacher Award in 2019.