Deep Transfer Learning: A Novel Collaborative Learning Model for Cyberattack Detection Systems in IoT Networks

Tran Viet Khoao, Dinh Thai Hoang, Senior Member, IEEE, Nguyen Linh Trung, Senior Member, IEEE, Cong T. Nguyen, Tran Thi Thuy Quynh, Member, IEEE, Diep N. Nguyen, Senior Member, IEEE, Nguyen Viet Ha, and Eryk Dutkiewicz, Senior Member, IEEE

Abstract—Federated learning (FL) has recently become an effective approach for cyberattack detection systems, especially in Internet of Things (IoT) networks. By distributing the learning process across IoT gateways, FL can improve learning efficiency, reduce communication overheads, and enhance privacy for cyberattack detection systems. However, one of the biggest challenges for deploying FL in IoT networks is the unavailability of labeled data and dissimilarity of data features for training. In this article, we propose a novel collaborative learning framework that leverages Transfer Learning (TL) to overcome these challenges. Particularly, we develop a novel collaborative learning approach that enables a target network with unlabeled data to effectively and quickly learn “knowledge” from a source network that possesses abundant labeled data. It is important that the state-of-the-art studies require the participated data sets of networks to have the same features, thus limiting the efficiency, flexibility, as well as scalability of intrusion detection systems. However, our proposed framework can address these problems by exchanging the learning knowledge among various deep learning (DL) models, even when their data sets have different features. Extensive experiments on recent real-world cybersecurity data sets show that the proposed framework can improve more than 40% as compared to the state-of-the-art DL-based approaches.

Index Terms—Cyberattack detection, cybersecurity, deep learning (DL), federated learning (FL), Internet of Things (IoT), transfer learning (TL).

I. INTRODUCTION

IN RECENT years, the rapid development of various technologies, such as 5G/6G, Industry 4.0, and Internet of Things (IoT), has enabled numerous applications to become an integral part in many aspects of our daily lives. However, such ever-fast growth has also led to an unprecedented massive amount of data and the proliferation of interconnected devices, e.g., sensors, smart cars, and cameras, which raises serious security and privacy concerns. Particularly, the increasing number of emerging applications has also brought forth many new types of cyberattacks. For example, the number of new (zero-day) cyberattacks has increased by 60% from 2018 to 2019 [1]. Besides the dire consequences to the economic, e.g., ransomware alone cost more than $5 billion globally in 2017 [2], cyberattacks pose serious threats to other areas with highly sensitive information, such as healthcare and public security. As a result, cyberattack detection methods play a key role in detecting and promptly preventing consequences of cyberattacks in future IoT networks.

Recently, with outstanding classification ability, machine learning (ML) techniques, especially deep learning (DL), have been widely applied for cyberattack detection problems. Particularly, DL models can effectively learn the signatures of various cyberattack types. Moreover, DL models even can detect new types of attacks that have never been learned/trained before [3]. Nevertheless, DL-based cyberattack detection systems are also facing some practical challenges. Particularly, conventional DL approaches usually require a huge amount of data to achieve a high performance. However, in many applications, data are very difficult to collect because they are often stored locally on user devices, such as IoT devices, smartphones, and wearable devices. This poses a threat to user privacy because sensitive data (e.g., location and private information) have to be sent over the network and stored at the centralized server for processing. Besides the privacy concerns, transmitting such a collectively large amount of data also imposes an extra communication burden...
over the network. Consequently, these limitations have been hindering the effectiveness of DL techniques in cyberattack detection systems.

To address these problems, federated learning (FL) has emerged to be a highly effective solution. Unlike conventional DL techniques that collect data and train the global model at a central server, FL enables the learning process to be distributed across all devices. Particularly, instead of sending data to a central server, the local data can be used to train a global model locally on each user device. Then, the obtained model weights of each device are periodically sent to a central server for aggregation. Afterward, the aggregated weights are sent back to all devices to update their local models’ weights. Since only the weights are transmitted in FL, both the privacy and communication overhead issues can be mitigated [4].

Despite its effectiveness, FL is still facing some challenges. Particularly, FL only performs well if the training data and the predicting data are independent and identically distributed (i.i.d.). Consequently, they are not robust to the changes in the system, e.g., changes in network traffic due to the mobility of users, new types of devices participating in the network, and so on. Moreover, the performance of FL largely relies on the availability of labeled data. However, acquiring sufficient labeled data might be costly and time-consuming. Even if the data are available, the participated user data usually have different structures such as features. This leads to difficulty or even mistakes when FL aggregates the global model. Consequently, they may not be suitable for the intensive training process of FL [4], [5].

To address these limitations, transfer learning (TL) has been emerging as a promising solution, especially for problems related to heterogeneous training data [6], [7], [8]. Unlike DL and FL techniques that are trained only for a specific problem, TL can utilize knowledge from rich resource data to enhance the training process and performance of the ML models. Particularly, by transferring knowledge from similar scenarios with a lot of high-quality data, TL can address the lack of labeled data for the target networks. Moreover, the TL can exchange knowledge even if the data features of the target and source networks are not very similar [6], [9]. However, if the data features are too different, TL might even make the learning process worse than that without using TL, i.e., negative transfer [6], [7], [8]. In the context of cyberattack detection for IoT networks, negative transfer might be a serious problem since different networks may have various types of devices generating different data.

In this article, we propose a novel collaborative learning framework that utilizes the strengths of both TL and FL to address the limitations of conventional DL-based cyberattack detection systems. Particularly, we consider a scenario with two different IoT networks. The first network (source network) has an abundant labeled data resource, while the second network (source network) has very little data resource (and most of them are unlabeled). Here, unlike most of the current works that assume that the data at these networks have the same features [10], we consider a much more practical and general case in which data at these two networks may have different features. To address the problem of dissimilar feature spaces of the target and source networks, we propose to transform them into a new joint feature space. In this case, at each learning round of the FL process, trained models of target and source networks can be exchanged through the joint feature space. Thus, by periodically exchanging and updating the trained model, the target network can eventually achieve the converged trained deep neural network (DNN) that can predict attacks with high accuracy (thanks to useful knowledge transferred from the source network). Besides the exchanging and updating the learning model iteratively, we use a small number of mutual samples between two networks to mitigate the negative TL. More importantly, unlike FL where networks try to train a joint global model, our proposed framework enables the participating networks to obtain their particular trained models that are specific to their networks, i.e., better predict attacks for particular networks with different data structures. Extensive experiments on recent real-world data sets, including N-BaIoT [11], [12], KDD [13], NSL-KDD [14], and UNSW [15] show that our proposed framework can achieve an accuracy of up to 99% and an improvement of up to 40% over the unsupervised learning approach. The main contributions of this article can be summarized as follows.

1) We propose a novel collaborative learning framework that can effectively detect cyberattacks in decentralized IoT systems. By combining the strengths of FL and TL, our proposed framework can improve learning efficiency and the accuracy of cyberattack detection in comparison with the conventional DL-based cyberattack detection systems.

2) We propose an effective TL approach that can allow the DL model from the rich-data network to transfer useful knowledge to the low-data network even they have different features for cyberattack detection in IoT networks.

3) We perform extensive experiments on recent real-world data sets including N-BaIoT, KDD, NSL-KDD, and UNSW to evaluate the performance of the proposed collaborative learning framework. The results show that our proposed approach can achieve an accuracy of up to 99% and an improvement of up to 40% over the unsupervised learning approach.

The remainder of this article is organized as follows. We first discuss related works in Section II. We then propose the federated TL model for cyberattack detection in Section III. After that, simulation settings and results are discussed in Section IV. Finally, we conclude this article in Section V.

II. RELATED WORK

A. Deep Learning for Cyberattack Detection

There have been a rich literature proposing DL approaches for cyberattack detection. In [16], a DNN model is developed to detect zero-day attacks based on two types of data, i.e., network activities and local system activities. The results show that for most of the data sets, the proposed DNN can
achieve a higher detection accuracy and lower false-positive rate compared to those of the other conventional ML classifiers, such as $K$-nearest neighbors (KNNs) and support vector machine (SVM). Another DL approach is proposed in [17] to detect cyberattacks in the mobile cloud computing environments. The main difference between [16] and [17] is that the approach in [17] consists of a feature analysis phase before the learning phase. In the analysis phase, the data sets are analyzed to identify meaningful features, thereby, reducing the data dimension and computational complexity. Experiments on the KDD [13], NSL-KDD [14], and the UNSW [15] data sets show that the proposed approach can achieve a detection accuracy of up to 97.1%.

B. Federated Learning for Cyberattack Detection

With the advent of FL, the research focus has recently shifted toward applying this framework for cyberattack detection, especially in environments with numerous devices, such as IoT and mobile edge networks. In [18], an FL framework is proposed for cyberattack detection in an edge network. In this network, the data for intrusion detection are stored locally at each edge node. The edge nodes train their data locally and send their models’ weights to an FL server for aggregating. After aggregation, the FL server sends the weights back to all edge nodes. In this way, each edge node can benefit from the other nodes’ data and training while protecting its privacy and reducing the network’s communication burden. Experiments with the NSL-KDD data sets show that the proposed approach can achieve an accuracy of up to 99.2%. Another FL approach is proposed in [19] for attack detection in industrial cyber-physical systems. In the considered setting, there are multiple cyber-physical systems acting as FL nodes. However, unlike the previous frameworks, the authors propose a novel architecture combining a convolution neural network (CNN) and a gated recurrent unit for training at each FL node. Experiments with self-collected data show that the proposed approach can outperform other state-of-the-art approaches, e.g., [20], [21], and [22], with an accuracy up to 99.2%. However, because of the limitations of FL as presented in the previous section, the learning model can only combine data with the same features and labels.

C. Transfer Learning for Cyberattack Detection

Although FL techniques can effectively address the privacy and communication load concerns of conventional ML for cyberattack detection, they are still facing some challenges. Particularly, FL approaches usually require high-quality and labeled data for training. However, collecting and labeling such data is expensive and time-consuming, especially for large-scale systems. On the other hand, unlabeled data are often abundant in environments, such as IoT and mobile edge networks. Thus, a deep TL approach is proposed for IoT intrusion detection in [23] based on network activities, which can utilize both labeled and unlabeled data. In this approach, the authors employ two AEs. The first AE is trained with labeled data, while the second AE is trained with unlabeled data. Then, the knowledge is transferred from the first AE to the second AE by minimizing the maximum mean discrepancy (MMD) distances between their weights. Experiments over nine IoT data sets were conducted to show that the proposed approach can achieve higher area under the curve (AUC) scores compared to those of several other approaches.

Besides analyzing network traffic, another approach to detect cyberattacks is to analyze the devices’ fingerprints. Particularly, attackers may try to impersonate a device in the system by copying its signal. For this kind of attack, ML techniques can be used to detect if the signals are coming from the real device or the malicious device. TL approaches, such as [24], [25], [26], and [27], are proposed to identify cyberattacks based on device fingerprints. Among them, [26] and [27] leverage the environmental effects to classify signals from devices. To improve the classification accuracy and address the lack of data, these approaches transfer the knowledge from nearby devices (since they share similar environmental effects). On the other hand, [24] and [25] leverage the knowledge from previous experiences, i.e., data collected in the past. These past data are then combined with the current data for training, thereby, addressing the lack of fingerprint data.

Unlike all the abovementioned approaches, the collaborative learning framework proposed in this article can leverage the strengths of both FL and TL to address limitations of ML-based intrusion detection systems, e.g., lack of labeled data, privacy, and heterogeneous data feature space. Moreover, in our approach, each IoT network has a separated model that is fine-tuned specifically for that network, therefore, the model is more effective for that network’s cyberattack detection compared to FL frameworks with a single model for all networks. Furthermore, our proposed system model can utilize knowledge from both source and target data in the network instead of only transferring knowledge from a single source as proposed in most of the mentioned TL frameworks [23], [24], [25], [28], thereby, mitigating the negative transfer problem.

III. PROPOSED FEDERATED TRANSFER LEARNING FRAMEWORK FOR CYBERATTACK DETECTION IN IoT NETWORKS

A. System Model

The conventional FL model requires to use a centralized server to maintain and aggregate all the trained models in the whole learning process. However, this may lead to a high cost to maintain and may not be effective to deploy in IoT networks. Thus, in this work, we propose a federated TL model that allows the learning process to be performed more flexibly and effectively in IoT environments. In particular, we consider a network which has unlabeled data (e.g., Network B as illustrated in Fig. 1), and it wants to learn more knowledge from other networks with abundant labeled data. In this case, this network will connect with a target network (e.g., Network A as illustrated in Fig. 1) and nominate itself as a centralized node which can train its own data as well as perform TL to exchange knowledge with the target network.

We denote a labeled cybersecurity data set $D_A = \{X_A, y_A, F_A\}$ of Network A with $(X_A, Y_A) = \{x_1^A, y_1^A, x_2^A, y_2^A, \ldots, x_{M_A}^A, y_{M_A}^A\}$ where $M_A$ is the number
KHOA et al.: DEEP TRANSFER LEARNING: A NOVEL COLLABORATIVE LEARNING MODEL

Fig. 1. Illustration of a system model for cyber attack detection in IoT networks.

of samples of data set A. In contrast, Network B has an unlabeled cybersecurity data set \(D_B = \{X^B, F^B\}\) with \((X^B) = \{x^B_1, x^B_2, \ldots, x^B_M\}\) where \(M_B\) is the number of samples of data set B. \(F^A\) and \(F^B\) are the feature spaces of Network A and Network B, respectively. The proposed model will perform TL between two neural network by minimizing the total loss \(J\) to predict the label \(P(z^B)\) for the unlabeled data set of Network B. In this way, the network can help to improve the accuracy in identifying network traffics by learning useful knowledge from other labeled networks. Each network can be managed by an IoT gateway and possesses its own private data set. The IoT gateway uses its DL model to detect normal and abnormal traffics. It is important to note that, unlike conventional FL approaches [29], in this work, we consider a practical scenario in which the data sets of networks may have different features.

B. Proposed Federated Transfer Learning Approach for Cyberattack Detection

In this section, we propose a highly effective federated TL model that can exchange knowledge between an unlabeled network and multiple networks which may have different features. To better analyze the impact of our proposed approach, we consider a specific scenario in which one labeled network is used as a source network to support an unlabeled network (i.e., target network). The scenario with one unlabeled network and multiple labeled networks can be straightforwardly extended, and we leave it for future study. Fig. 2 describes the training and predicting processes of the FTL algorithm that we use in this case. The table of notations is presented in Table I. As described in previous section, Network A and Network B have their data set \(D^A\) and \(D^B\), respectively. They also have their model parameters called \(W^A\) and \(W^B\). The outputs of two neural networks are calculated as follows:

\[
Z^A = W^A \ast X^A \quad (1a)
\]
\[
Z^B = W^B \ast X^B. \quad (1b)
\]

We need to find the prediction function \(P(z^B_i) = P(z^B_i, y^A_1, \ldots, y^A_M, y^B_{jL}, y^B_{jR})\) to predict the output of Network B. To find a high-quality predict function, we first need to minimize the loss function using the labeled data set as follows:

\[
\arg \min_{W^A, W^B} J^B = \sum_{i} J^B(y^A_i, P(z^B_i)) \quad (2)
\]

where \(M_c\) is the number of predicted labels, and \(J^B\) represents the loss of the loss function which depends on the type of output or mechanism, i.e., the logistic loss function [30] with
the predicted value $z$ and the labeled $y$

$$j^B(z, y) = \log(1 + \exp(-z \times y)).$$

(3)

In addition, data sets A and B may have some overlapping samples, and thus, we can use these samples to optimize the loss function. We denote $M_{AB}$ as the overlapping samples between data set A and data set B. We need to minimize the alignment loss function between A and B as follows:

$$\arg\min_{W^A, W^B} J^{AB} = \sum_{i} J^{AB}(z_i^A, z_i^B)$$

(4)

where $J^{AB}$ represents the alignment loss function. The common alignment loss function can be represented in modulus $j^{AB} = ||z_i^A - z_i^B||^2$ or angle $j^{AB} = -z_i^A \cdot z_i^B$. Last, we add the regularization $J^R_A = \sum_{l=1}^{L_A} ||w_{l}^A||^2$ and $J^R_B = \sum_{l=1}^{L_B} ||w_{l}^B||^2$ in which $L_A$ and $L_B$ are the numbers of layers in neuron Network A and Network B, respectively, to find the final loss function that needs to be minimized

$$\arg\min_{W^A, W^B} J = J^R + \gamma J^{AB} + \frac{\lambda}{2} (J^R_A + J^R_B)$$

(5)

where $\gamma$ and $\lambda$ are the weight parameters. The gradient for updating $W^A$ and $W^B$ are calculated by the following formula:

$$\frac{\partial J}{\partial w_{l}^i} = \frac{\partial J^B}{\partial w_{l}^i} + \gamma \frac{\partial J^{AB}}{\partial w_{l}^i} + \lambda w_{l}^i.$$  

(6)

The training process is presented in Algorithm 1. Specifically, we first initialize $W^A$ and $W^B$. Next, we calculate $z_i^A$ and $z_i^B$ from the input samples of data set A ($D^A$) and data set B ($D^B$) as shown in (1). Then, Network A sends $\{z_i^A, y_i^A\}$ to Network B to calculate $J^B$, the alignment loss function $J^{AB}$ and the gradients of $J^B$ as shown in (2), (4), (5), and (6), respectively. Similarly, Network B sends $\{z_i^B\}$ to Network A to calculate $J^A$ as in (5). In (4), we use $M_{AB}$ as the mutual samples of two data sets. For example, the same IoT devices are attacked by the same types of cyberattacks in different networks. Each network extracts the attack data with different features, e.g., Network A uses timeslot, packet header, and IP address while Network B uses MAC address, error packets, and frame header. The number of mutual samples is an important factor that strongly supports the learning process between two networks (we will explain it more details in Section IV). After that, we calculate the final loss function $J$ and the gradient as in (5) and (6). Finally, Network A and Network B update their model parameters based on the gradient and loss functions. This process continuously repeats until the system converges or reaches the maximum number of iterations to minimize the final loss function in (5).

When the training completes, the prediction process described in Algorithm 2 is called to predict the final result of the unlabeled data set $D_B$. In this process, both Network A and Network B have their trained models. Similar to the training process, the data set $D_B$ first goes through the trained model of Network B to calculate $Z^B$. Then, Network B sends $Z^B$ to Network A to archive the TL knowledge from the trained model of Network A. Network A predicts the results and sends them back to Network B to classify the attack and normal behaviors of the network.
Algorithm 1 Federated TL Algorithm: Training Process

1. **Input:** The learning rate \( \eta \), the weight parameter \( \gamma, \lambda \), the maximum iteration \( T \), the tolerance \( t \) and Network A and Network B initialize model parameters \( W^A, W^B \).
2. **Output:** The trained model parameter \( W^A, W^B \).
3. \( \text{iteration} = 0 \)
4. **while** \( \text{iteration} \leq T \) **do**
5. \( \text{Network A performs:} \)
6. \( z^A_i = h^A_i \ast x^A_i \) for \( i \in D_A \);
7. \( \text{Send } \{z^A_i, y^A_i\} \text{ to Network B;} \)
8. \( \text{Network B performs:} \)
9. \( z^B_i = h^B_i \ast x^B_i \) for \( i \in D_B \);
10. \( \text{Send } \{z^B_i\} \text{ to Network A;} \)
11. \( \text{Network A performs:} \)
12. \( \text{Compute } J^{AB} \) and \( J^B \), then send them to Network B;
13. \( \text{Network B performs:} \)
14. \( \text{Compute } \frac{\partial J^B}{\partial w^B} \) and \( J^A \) and \( J^{AB} \), then send them to Network A;
15. \( \text{Network A performs:} \)
16. \( \text{Update } w^A_i = w^A_i - \eta \frac{\partial J^A}{\partial w^A_i}; \)
17. \( \text{Network B performs:} \)
18. \( \text{Update } w^B_i = w^B_i - \eta \frac{\partial J^B}{\partial w^B_i}; \)
19. **if** \( J_{prev} - J \leq t \) **then**
20. \( \text{Send stop signal to Network B;} \)
21. \( \text{Break.} \)
22. **else**
23. \( J_{prev} = J; \)
24. \( \text{iteration} = \text{iteration} + 1; \)
25. \( \text{continue;} \)
26. **end if**
27. **end while**

Algorithm 2 Federated TL Algorithm: Predicting Process

1. **Input:** The model parameters \( W^A, W^B \) and dataset \( X_B \);
2. **Output:** The prediction \( Y^B \);
3. \( \text{Network B performs:} \)
4. \( z^B_i = h^B_i \ast x^B_i \) for \( i \in D_B \);
5. \( \text{Send } \{z^B_i\} \text{ to Network A;} \)
6. \( \text{Network A performs:} \)
7. \( \text{Compute } P(z^B_i) = W^A[z^B_i] \) and send it to Network B.

C. Evaluation Methods

As mentioned in [31] and [32], the confusion matrix is typically used to evaluate system performance, especially for intrusion detection systems. We denote TP, TN, FP, and FN to be “True Positive,” “True Negative,” “False Positive,” and “False Negative,” respectively. The receiver operator characteristic (ROC) is created by plotting the TPR over FPR at different thresholds. Then, we use AUC to evaluate the performance of the algorithm in the following formula:

\[
\xi = \int_{0}^{1} \text{TP} \left( \text{FP}^{-1}(x) \right) dx. \tag{7}
\]

In our experiments, we randomly select samples from original data set to test the algorithm. In this scenario, the \( p \)-value is often used to evaluate the results of random tests, and is given by

\[
p = F(\xi | \mu, \sigma) = \frac{1}{\sqrt{2\pi} \sigma} \int_{-\infty}^{\xi} e^{-\frac{(\xi - \mu)^2}{2\sigma^2}} d\xi \tag{8}
\]

in which \( \mu \) is the mean and \( \sigma \) is the standard deviation. The results are calculated by the significant number with the following formula:

\[
\text{Sig} = F^{-1}(p | \mu, \sigma) = \{ \xi : F(\xi | \mu, \sigma) = p \} \tag{9}
\]

where \( \text{Sig} \) is the significant number that represents the results of 30 random runs and the confidence of this number is calculated by \( \text{conf} = 1 - p \). In a normal situation, \( p \) is considered confidence, when it has values around 0.01 and 0.05, corresponding to the confidence of significant numbers is around 99% and 95%.

IV. PERFORMANCE ANALYSIS

A. Data Sets

In this experiment, we use four popular cybersecurity data sets, namely, the N-BaIoT [11], [12], KDD [13], NSL-KDD [14], and UNSW [15] data sets to evaluate the performance of the proposed method. The Network-based Detection of IoT Botnet Attacks (N-BaIoT) data set [11], [12] includes the information collected in the setup network about the normal and attack situation. The attack was performed by servers to nine IoT devices and the total network behavior was captured by the sniffer server to extract data set. This data set is characterized by 115 features for both normal and attack behaviors. In this data set, the attack type is the Distributed Denial of Service (DDoS) which was implemented by two well-known botnets, namely, Mirai and BASELITE. The BASELITE botnet includes five types of attacks, i.e., network scanning (scan), spam data sending (junk), UDP flooding (udp), TCP flooding (tcp), and the join of sending spam data and opening port to specific IP address (combo). Besides BASELITE, the Mirai botnet also includes five types of attacks, i.e., scan, ACK flooding (ack), SYN flooding (syn), udp, and optimized UDP flooding (udplan).

In addition to IoT data sets, we also want to evaluate our proposed solution on some classical intrusion detection data sets, i.e., KDD [13], NSL-KDD [14], and UNSW [15] data sets. The KDD data set [13] includes many different kinds of network attacks simulated in the military network environment. The KDD data set has 41 features and it classifies attacks into 4 groups, including Denial of Service (DoS), probe, user to root (U2R), and remote to local (R2L). The NSL-KDD data set [14] inherits the properties from KDD [13] data set, such as the features and types of attacks but eliminates the redundant samples in the training data set and the duplicated samples in the testing data set. Although both KDD and NSL-KDD data sets are well-known and used in many research works, they were developed long time ago. Thus, some modern attacks were not involved. Therefore, a recent data set, i.e., UNSW data set [15], is considered in this work. Unlike KDD and NSL-KDD, the feature space of this data set includes 42 types and 9 kinds of attacks, namely, DoS, backdoors,
TABLE II
DATA SET PREPARATION

| Dataset | Device name                        | Features of Network A | Features of Network B | Total features |
|---------|------------------------------------|-----------------------|-----------------------|----------------|
| IoT1    | Danmini_Doorbell                    | 85                    | 30                    | 115            |
| IoT2    | Ecobee_Thermostat                  | 85                    | 30                    | 115            |
| IoT3    | Ennio_Doorbell                      | 85                    | 30                    | 115            |
| IoT4    | Philips_B120N10_Baby_Monitor        | 85                    | 30                    | 115            |
| IoT5    | Provision_PT_737E_Security_Camera   | 85                    | 30                    | 115            |
| IoT6    | Provision_PT_838_Security_Camera    | 85                    | 30                    | 115            |
| IoT7    | Samsung_SNI_1011_N_Webcam           | 85                    | 30                    | 115            |
| IoT8    | SimpleHome_XCS7_1002_WHT_Security_Camera | 85               | 30                    | 115            |
| IoT9    | SimpleHome_XCS7_1003_WHT_Security_Camera | 85                | 30                    | 115            |
| KDD     |                                    | 31                    | 10                    | 41             |
| NSLKDD  |                                    | 31                    | 10                    | 41             |
| UNSW    |                                    | 31                    | 11                    | 42             |

TABLE III
RESULTS WITH MULTIPLE DATA SETS IN CASE 1. (a) RESULTS WITH p = 1. (b) RESULTS WITH p = 3. (c) RESULTS WITH p = 5

(a) FTL UDL

|        | FTL   | UDL |
|--------|-------|-----|
| IoT1   | 85.771| 45.753 |
| IoT2   | 83.795| 63.171 |
| IoT3   | 94.286| 80.453 |
| IoT4   | 79.241| 77.885 |
| IoT5   | 90.605| 81.876 |
| IoT6   | 91.179| 82.703 |
| IoT7   | 90.670| 85.183 |
| IoT8   | 82.960| 65.256 |
| IoT9   | 83.222| 73.072 |
| KDD    | 99.315| 80.477 |
| NSLKDD | 98.485| 83.025 |
| UNSW   | 97.072| 68.449 |

(b) FTL UDL

|        | FTL   | UDL |
|--------|-------|-----|
| IoT1   | 87.398| 49.770 |
| IoT2   | 85.672| 65.793 |
| IoT3   | 94.896| 81.070 |
| IoT4   | 81.672| 77.885 |
| IoT5   | 91.517| 82.013 |
| IoT6   | 92.059| 82.703 |
| IoT7   | 92.030| 86.013 |
| IoT8   | 85.197| 68.161 |
| IoT9   | 85.072| 73.078 |
| KDD    | 99.395| 81.304 |
| NSLKDD | 98.534| 83.450 |
| UNSW   | 97.141| 69.124 |

(c) FTL UDL

|        | FTL   | UDL |
|--------|-------|-----|
| IoT1   | 88.259| 51.897 |
| IoT2   | 86.666| 67.181 |
| IoT3   | 95.220| 81.397 |
| IoT4   | 82.959| 77.885 |
| IoT5   | 92.000| 82.085 |
| IoT6   | 92.525| 82.703 |
| IoT7   | 92.750| 86.453 |
| IoT8   | 86.381| 69.700 |
| IoT9   | 86.052| 73.082 |
| KDD    | 99.438| 81.742 |
| NSLKDD | 98.561| 83.675 |
| UNSW   | 97.177| 69.482 |

![Fig. 3. Data of participated networks used in this experiment.](image)

worms, fuzzers, analysis, reconnaissance, exploits, shellcode, and generic.

**B. Experiment Setup**

In this section, we carry out experiments using all the aforementioned data sets to evaluate the performance of the proposed solution. In this experiment, we denote IoT1-9 as the data set names of nine IoT devices. Table II describes the total features and the representative names of data sets that we use in this experiment. Fig. 3 also describes the separated data in each data set in this experiment. In this experiment, the participated data are randomly selected from the data set. Then, the selected data are separated into label data (data of Network A) and unlabeled data (data of Network B) with different features as described in Table II. These data have about 10% mutual samples of total data set samples. We experiment with two cases, i.e., the first one is with 2000 unlabeled data and 9577 labeled data (CASE 1), the second one is with 10 000 unlabeled data and 47 893 labeled data (CASE 2).

In this setup, we consider a baseline solution with the state-of-the-art unsupervised DL (UDL) model which clusters the unlabeled data into normal and attack behaviors based on autoencoder and k-means techniques [33]. The UDL model includes an autoencoder and KNN to cluster the unlabeled data. In addition, we consider the second baseline solution that uses both supervised and unsupervised data sets to feed the FTL learning models. The FTL will exchange the knowledge from the supervised learning model and the unsupervised learning model to improve the accuracy of learning as well as increase the precise of identifying attack and normal behaviors of the unlabeled data. Then, we measure the AUC of this process 30 times to calculate the signification number of the
C. Experimental Results

In this section, we show the results of our experiments with different kinds of cybersecurity data sets.

1) Accuracy Comparison: In this section, we compare the performance of FTL and the UDL method in terms of the significant number of each $p$ as explained in Section III. Tables III and IV describe the significant number of each data set with $p = 1$, $3$, and $5$ corresponding to the confidence of 99%, 97%, and 95%.

In general, Tables III and IV show that the significant numbers of all data sets increase as $p$ increases. This is because in (9), we calculate the significant number based on a series of 30 continuous AUC results. When $p$ increases, the AUC results increase in all tables. This demonstrates that most of the AUC results in 30 series are higher than the significant number in the case where $p = 1$.

Table III(c) shows the significant numbers of participated data sets with $p = 5$ in CASE 1. In this table, the IoT1 and UNSW data sets show a significant gap of about 30% and 40% between FTL and UDL. These results show the difficulty of clustering in recognizing the groups of samples and the advantage of collaborative learning in these data sets. The other ten data sets have gaps of around 10%–20% between the two methods, which demonstrate the stability of our proposed solution for any cybersecurity data set.

In addition, Table IV(c) shows the significant numbers of multiple data sets with $p = 5$ in CASE 2. In this table, the significant numbers also have a gap of around 10%–40% between the two solutions. It shows the common trend that the significant numbers increase for most data sets when the number of samples increases. However, in IoT2, IoT5, and IoT6 data sets, the significant numbers slightly decrease because of the randomly selected samples from the original data set. It also can be demonstrated by the high fluctuation of the reconstruction errors of IoT2, IoT5, and IoT6 data sets in Fig. 5(b) compared with other data sets. However, in all studied data sets,
our proposed solution still performs much better than the state-of-the-art UDL solution. These results demonstrate that our solution can work efficiently in all IoT and conventional cybersecurity data sets in detecting cyberattacks in the network.

2) Reconstruction Error Analysis: In this section, we discuss the convergence of the FTL algorithm in each data set. Fig. 4 describes the reconstruction errors of the nine IoT data sets and the conventional data sets like KDD, NSLKDD, and UNSW in CASE 1. Fig. 5 describes the reconstruction errors of study data sets in CASE 2.

In Figs. 4(a) and 5(a), we can see that at the first few epochs, the errors are very high for KDD (up to $2.6 \times 10^5$ in CASE 1 and $12 \times 10^5$ in CASE 2), but this error dramatically reduces to $0.3 \times 10^5$ in CASE 1 and $1.5 \times 10^5$ in CASE 2 after only 200 epochs. For NSLKDD and UNSW, they have very similar trends with $0.75 \times 10^5$ in CASE 1 and $3.8 \times 10^5$ in CASE 2 at the beginning and gradually reduce to $0.4 \times 10^5$ in CASE 1 and $1.9 \times 10^5$ in CASE 2 after 200 epochs, respectively. After 200 epochs, the algorithm converges as all the reconstruction error curves are flattened.

Figs. 4(b) and 5(b) show the reconstruction errors of nine IoT data sets in both CASE 1 and CASE 2. We can observe the same trend overall data sets, i.e., all errors gradually reduce when the number of epochs increases. However, it can be observed that the trend exhibits some fluctuations in comparison with the trends in Figs. 4(a) and 5(a) because of the heterogeneous distribution in IoT data sets. The high fluctuation of the reconstruction errors of IoT2, IoT5, and IoT6 data sets in Fig. 5(b) also explains why their significant numbers reduce when the number of samples increases in CASE 2. However, the reconstruction errors of all studied data sets in our proposed solution dramatically decrease and become stable after 200 running epochs in both cases.

3) Mutual Information Analysis: As mentioned in the previous section, Network A and Network B may share a number of mutual samples. The FTL algorithm exploits the information of these mutual samples to perform the
prediction for unlabeled data of Network B. This section provides the analysis results to identify how this mutual information can affect to the results of label prediction. In this section, we perform the simulation in CASE 2 with a larger number of samples than in CASE 1. Fig. 6 gives information about the variation of AUC when the percentage of mutual data increases.

Fig. 6(a) shows the increase of AUC on KDD, NSLKDD, and UNSW data sets when the percentage of mutual samples increases from 0.005% to 10%. The AUC of KDD and UNSW data sets sharply increase and remain stable at around 96% on the NSLKDD data set with about 5% to 10% mutual samples. A similar trend happens with the IoT data sets in Fig. 6(b) when the AUCs of all nine IoT data sets increase and remain stable at around 96% on the number of samples than in CASE 1. Fig. 6 gives information for unlabeled data of Network B. This section provides the analysis results to identify how this mutual information can affect to the results of label prediction.

In summary, the results with 12 cybersecurity data sets show the outperformance of our proposed model in comparison with the state-of-the-art UDL in term of accuracy as shown in Table III for CASE 1 and Table IV for CASE 2, especially with IoT and UNSW data sets. Moreover, the reconstruction errors show a fluctuation of the IoT data sets when the number of samples increases due to noise from the collected data sets of some IoT devices. Finally, we vary the amount of mutual data between two networks to evaluate the accuracy of our proposed model. The results show that the proposed model can achieve high performance with 10% mutual data with all data sets.

V. CONCLUSION

In this work, we have proposed a novel collaborative learning framework to address the limitations of current ML-based cyberattack detection systems in IoT networks. In particular, by extracting and transferring knowledge from a network with abundant labeled data (source network), the intrusion detection performance of the target network can be significantly improved (even if the target has very few labeled data). More importantly, unlike most of the current works in this area, our proposed framework can enable the source network to transfer the knowledge to the target network even when they are different data structures, e.g., different features. The experimental results then show that the accuracy of prediction of our proposed framework is significantly improved in comparison with the state-of-the-art UDL model. In addition, the convergence of the proposed collaborative learning model is also analyzed with various cybersecurity data sets. In future work, we can consider using other effective TL techniques to make TL processes more stable and achieve better performance, especially when the amount of mutual information is very limited.

REFERENCES

[1] Y. Keshet, “Half of the malware detected in 2019 was classified as zero-day threats, making it the most common malware to date,” Mar. 2020. [Online]. Available: https://www.cynet.com/blog/half-of-the-malware-detected-in-2019-was-classified-as-zero-day-threats-making-it-the-most-common-malware-to-date/

[2] S. Morgan, “Global Ransomware damage costs predicted to hit $11.5 billion by 2019,” Mar. 2021. [Online]. Available: https://cybersecurityventures.com/ransomware-damage-report-2017-part-2/

[3] T. V. Khoa et al., “Collaborative learning for Cyberattack detection in blockchain networks,” Mar. 2022, arXiv:2203.11076.

[4] W. Y. B. Lim et al., “Federated learning in mobile edge networks: A comprehensive survey,” IEEE Commun. Surveys Tuts., vol. 22, no. 3, pp. 2031–2063, 3rd Quart, 2020.

[5] M. Aledhari, R. Razzaq, R. M. Parizi, and F. Saeed, “Federated learning: A survey on enabling technologies, protocols, and applications,” IEEE Access, vol. 8, pp. 140699–140725, Jul. 2020.

[6] C. T. Nguyen et al., “Transfer learning for future wireless networks: A comprehensive survey,” Feb. 2021, arXiv:2102.07572.

[7] S. Niu, Y. Liu, J. Wang, and H. Song, “A decade survey of transfer learning (2010–2020),” IEEE Trans. Artif. Intell., vol. 1, no. 2, pp. 151–166, Oct. 2020.

[8] F. Zhuang et al., “A comprehensive survey on transfer learning,” Proc. IEEE, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[9] M. Xu, D. T. Hoang, J. Kang, D. Niyato, Q. Yan, and D. I. Kim, “Secure and reliable transfer learning framework for 6G-enabled Internet of Vehicles,” IEEE Wireless Commun., early access, May 9, 2022, doi: 10.1109/MWC.021.200542.

[10] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke, and L. Shu, “Federated deep learning for cyber security in the Internet of Things: Concepts, applications, and experimental analysis,” IEEE Access, vol. 9, pp. 138509–138542, Oct. 2021.

[11] Y. Meidan et al., “N-BaIoT-network-based detection of IoT botnet attacks using deep autoencoders,” IEEE Pervasive Comput., vol. 17, no. 3, pp. 12–22, Jul–Sep. 2018.

[12] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, “Kitsune: An ensemble of autoencoders for Online network intrusion detection,” Feb. 2018, arXiv:1802.06989.

[13] “KDD dataset.” Assessed: Feb. 8, 2021. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/

[14] “NSL-KDD dataset.” Assessed: Feb. 8, 2021. [Online]. Available: https://www.unb.ca/cic/datasets/NSL.html

[15] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in Proc. Mil. Commun. Inf. Syst. Conf., Nov. 2015, pp. 1–6.

[16] R. Vinyakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, “Deep learning approach for intelligent intrusion detection system,” IEEE Access, vol. 7, pp. 41525–41550, Apr. 2019.

[17] K. K. Nguyen, D. T. Hoang, D. Niyato, P. Wang, D. Nguyen, and E. Dutkiewicz, “Cyberattack detection in mobile cloud computing: A deep learning approach,” in Proc. IEEE Wireless Commun. Netw. Conf., Apr. 2018, pp. 1–6.

[18] A. Abeshu and N. Chilamkurti, “Deep learning: The frontier for distributed attack detection in fog-to-things computing,” IEEE Commun. Mag., vol. 56, no. 2, pp. 169–175, Feb. 2018.

[19] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, “DeepFed: Federated deep learning for intrusion detection in industrial cyber-physical systems,” IEEE Trans. Ind. Informat., vol. 17, no. 8, pp. 5615–5624, Aug. 2021.

[20] D. T. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. R. Sadeghi, “DioT: A federated self-learning anomaly detection system for IoT,” in Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst., Jul. 2019, pp. 756–767.

[21] W. Schneele and G. Thamilarasu, “Attack detection using federated learning in medical cyber-physical systems,” in Proc. 28th Int. Conf. Comput. Commun. Netw., Jun. 2019, pp. 1–8.

[22] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, “FedHealth: A federated transfer learning framework for wearable healthcare,” IEEE Intell. Syst., vol. 35, no. 4, pp. 83–93, Jul-Aug. 2020.

[23] L. Yu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, “Deep transfer learning for IoT attack detection,” IEEE Access, vol. 8, pp. 107335–107344, Jun. 2020.

[24] Y. Sharaf-Dabagh and W. Saad, “Transfer learning for device fingerprinting with application to cognitive radio networks,” in Proc. IEEE 26th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun., Aug. 2015, pp. 2138–2142.

[25] C. Zhao, Z. Cai, M. Huang, M. Shi, X. Du, and M. Guizani, “The identification of secular variation in IoT based on transfer learning,” in Proc. Int. Conf. Comput. Netw. Commun., Mar. 2018, pp. 878–882.

[26] Y. Sharaf-Dabagh and W. Saad, “On the authentication of devices in the Internet of Things,” in Proc. IEEE 17th Int. Symp. World Wireless Mobile Multimedia Netw., Jun. 2016, pp. 1–3.
Tran Viet Khoa received the B.Sc. degree in electronics and telecommunications from the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, in 2008, and the M.Sc. degree from Paris-Sud 11, Orsay, France, in 2010. He is currently pursuing the Ph.D. degree with UTS-VNU Joint Technology and Innovation Research Centre between Vietnam National University and the University of Technology Sydney, Sydney, NSW, Australia. He was a Network Engineer with Viettel Network Corporation, Hanoi, from 2012 to 2018. His research interests include cyberattack detection, IoT, deep learning, and blockchain technology.

Dinh Thai Hoang (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2016. He is currently a Faculty Member with the School of Electrical and Data Engineering, University of Technology Sydney, Sydney, NSW, Australia. His research interests include emerging topics in wireless communications and networking, such as machine learning, edge intelligence, cybersecurity, IoT, and Metaverse.

Dr. Hoang has received several awards, including the Australian Research Council and the IEEE TCSC Award for Excellence in Scalable Computing (Early Career Researcher). He is currently an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and an Associate Editor of IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.

Nguyen Linh Trung (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in electrical engineering from Queensland University of Technology, Brisbane, QLD, Australia, in 1996 and 2005, respectively. Since 2006, he has been on the faculty with the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, where he was appointed as an Associate Professor of Electronic Engineering with the Faculty of Electronics and Telecommunications in 2012. Application-wise, his research interests include epilepsy and Alzheimer’s disease, communication channel estimation, and IoT cybersecurity. His current research interests include subspace and tensor-based methods for signal and information processing.

Dr. Trung is currently chairing the IEEE Signal Processing Society—Vietnam Chapter. From 2017 to 2020, he had been the Technical Editor-in-Chief of the Journal of Research and Development on Information and Communication Technology in Vietnam.

Cong T. Nguyen received the B.E. degree in electrical engineering and information from Frankfurt University of Applied Sciences, Frankfurt, Germany, in 2014, and the M.Sc. degree in global production engineering and management from the Technical University of Berlin, Berlin, Germany, in 2016. He is currently pursuing the Ph.D. degree with UTS-HCMUT Joint Technology and Innovation Research Centre between the Ho Chi Minh University of Technology, Ho Chi Minh City, Vietnam, and the University of Technology Sydney, Sydney, NSW, Australia. His research areas include operations research, blockchain technology, game theory, and optimizations.

Tran Thi Thuy Quynh (Member, IEEE) was born in 1979. She received the B.Sc., M.Sc., and Ph.D. degrees in telecommunication engineering from the University of Engineering and Technology, Vietnam National University (VNU-UET), Hanoi, Vietnam, in 2001, 2005, and 2016, respectively.

Since 2009, she has been a Researcher with the Faculty of Electronics and Telecommunications, VNU-UET. Her research interests include microwave component and antenna design, applying signal processing methods for antenna arrays, and currently focusing on implementing test beds for networking and cybersecurity.

Diep N. Nguyen (Senior Member, IEEE) received the M.E. degree in electrical and computer engineering from the University of California at San Diego (UCSD), La Jolla, CA, USA, in 2008, and the Ph.D. degree in electrical and computer engineering from The University of Arizona (UA), Tucson, AZ, USA, in 2013.

He is currently a Faculty Member with the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Sydney, NSW, Australia. Before joining UTS, he was a DECRA Research Fellow with Macquarie University, Sydney, and a member of Technical Staff with Broadcom Corporation, Irvine, CA, USA, and ARCON Corporation, Boston, MA, USA, and consulting the Federal Administration of Aviation, Washington, DC, USA, on turning detection of UAVs and aircraft, and the U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base, OH, USA, on anti-jamming. His research interests include computer networking, wireless communications, and machine learning application, with emphasis on systems’ performance and security/privacy.

Dr. Nguyen received several awards from LG Electronics, UCSD, UA, the U.S. National Science Foundation, and the Australian Research Council. He is currently an Editor, an Associate Editor, and a Guest Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE ACCESS, IEEE SENSORS JOURNAL, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, and SCIENTIFIC REPORTS (Nature).
Nguyen Viet Ha was born in 1974. He received the B.Sc., M.Sc., and Ph.D. degrees in informatics from Takushoku University, Tokyo, Japan, in 1997, 1999, and 2002, respectively.

Since 2002, he has been on the faculty with the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, where he was appointed as an Associate Professor of Informatics with the Faculty of Information Technology in 2009. His current research interests include machine learning and natural language processing.

Eryk Dutkiewicz (Senior Member, IEEE) received the B.E. degree in electrical and electronic engineering and the M.Sc. degree in applied mathematics from the University of Adelaide, Adelaide, SA, Australia, in 1988 and 1992, respectively, and the Ph.D. degree in telecommunications from the University of Wollongong, Wollongong, NSW, Australia, in 1996.

His industry experience includes management of the Wireless Research Laboratory, Motorola, Sydney, NSW, Australia, in the early 2000's. He is currently the Head of the School of Electrical and Data Engineering, University of Technology Sydney, Sydney. He also holds a professorial appointment with Hokkaido University, Sapporo, Japan. His current research interests cover 5G/6G and IoT networks.