Semi-Supervised Spatiotemporal Deep Learning for Intrusions Detection in IoT Networks

Mohamed Abdel-Basset, *Senior Member, IEEE*, Hossam Hawash, Ripon K. Chakrabortty, *Member, IEEE*, and Michael J. Ryan, *Senior Member, IEEE*

***Abstract*—The rapid growth of the Internet of Things (IoT) technologies has generated a huge amount of traffic that can be exploited for detecting intrusions through IoT networks. Despite the great effort made in annotating IoT traffic records, the number of labeled records is still very small, increasing the difficulty in recognizing attacks and intrusions. This study introduces a semi-supervised deep learning approach for intrusion detection (SS-Deep-ID), in which we propose a multiscale residual temporal convolutional (MS-Res) module to finetune the network capability in learning spatiotemporal representations. An improved traffic attention (TA) mechanism is introduced to estimate the importance score that helps the model to concentrate on important information during learning. Furthermore, a hierarchical semi-supervised training method is introduced which takes into account the sequential characteristics of the IoT traffic data during training. The proposed SS-Deep-ID is easily integrated into a fog-enabled IoT network to offer efficient real-time intrusion detection. Finally, empirical evaluations on two recent data sets (CIC-IDS2017 and CIC-IDS2018) demonstrate that SS-Deep-ID improves the efficiency of intrusion detection and increases the robustness of performance while maintaining computational efficiency.**

***Index Terms*—Deep learning (DL), Internet of Things (IoT), semi-supervised learning.**

I. Introduction

THE RAPID advance of the Internet of Things (IoT) technologies [1], facilitated the development of a wide variety of innovative services and applications, such as smart manufacturing, smart healthcare, and smart transportation. Huge amounts of IoT traffic is communicated between various IoT entities [2] to pass information—such as switch control, smart device administration, communication details, and equipment maintenance details—which usually contains a variety of issues engendered by manufactured or normal irregularities [3], which can damage IoT communication. Consequently, operative intrusion detection has emerged as a necessary function in IoT networks [4].

Intrusion detection systems (IDSs) have been introduced to recognize intrusions that have avoided security techniques, to provide a vitally important second level of resistance in securing IoT networks [5]. IDS based on anomaly detection build a normal behavior profile and classify behaviors as attacks if they do not match this normal profile [6]. In view of this, IDSs can be divided into two categories: 1) signature-based IDS (S-IDS) and 2) anomaly-based IDS (A-IDS). The S-IDS aims to recognize spiteful interventions by investigating the correlation with previously learned signatures of recognized attacks [7]. However, since S-IDS are unable to detect new unseen intrusions, the workload of S-IDS increases by increasing the count of newly recognized intrusions that increase the number of signatures, hence limit its responsivity [5]. Additionally, S-IDS often necessitates intervention from human experts to inspect and analyze the signatures of novel attacks. On the other hand, A-IDS can distinguish unidentified outbreaks, which often exist in most kinds of IoT systems.

The difficulty in recognizing unknown intrusions is a particular issue for IoT networks which connect a wide range of devices with different computation resources, communication technologies, battery capacity, software, and operating systems. This heterogeneous nature challenges the deployment of security solutions and increases the attack surfaces, leasing to IoT networks being more prone to novel and unfamiliar intrusions [7]–[9]. Conventional machine learning (ML) techniques [13] have been shown to efficiently recognize important patterns in IoT traffic, hence proficiently identify cyber-attacks [6]. Nevertheless, ML has also been demonstrated to fail to scale for huge data sets (i.e., millions of records with over a hundred features) and have also been shown to realize unsatisfactory performance in detecting intrusions/cyber-attacks in the situation where the IoT nodes are extremely distributed [10], [11]. Alternatively, continuous improvements in deep learning (DL) techniques stimulate new IDS that are well equipped to treat and handle the required intrusions/cyber-attacks, degree of difficulty and complexity, and levels of distributivity [12]. This article, therefore, proposes a new A-IDS using the new DL model.

The situational information of the IoT network is aggregated rapidly over time, such that manual labeling of a large volume of IoT records becomes difficult, if not impossible, task. Nevertheless, a small portion of IoT traffics could be labeled and the remaining large portion can be kept unlabeled [14]. The semi-supervised DL approaches are ideally suited for such a situation and have been proven to be an effective model for classification and prediction [15], which enables the development of intelligent systems that can learn from huge

Manuscript received December 23, 2020; revised January 22, 2021; accepted February 17, 2021. Date of publication February 19, 2021; date of current version July 23, 2021. *(Corresponding author: Ripon K. Chakrabortty.)*

Mohamed Abdel-Basset and Hossam Hawash are with the Department of Computer Science, Zagazig University, Zagazig 44519, Egypt (e-mail: mohamed.abdelbasset@fci.zu.edu.eg; hossamreda@zu.edu.eg).

Ripon K. Chakrabortty and Michael J. Ryan are with the Capability Systems Centre, School of Engineering and IT, UNSW Canberra, ACT 2600, Australia (e-mail: r.chakrabortty@adfa.edu.au; mike.ryan@ieee.org).

Digital Object Identifier 10.1109/JIOT.2021.3060878

amounts of unlabeled data using a small number of labeled instances.

Since IoT records are engendered in sequential order, they can be dealt with as time-series data. Recurrent neural networks (RNN) have been investigated for this kind of data and shown outstanding performance. Among them, the long short-term memory (LSTM) and gated recurrent unit (GRU) emerged as an enhanced version of RNN for various sequential data applications [7]. The efficiency of the RNN has been validated in various intrusion/attack recognition studies [5], [7]. More recently, convolution neural networks (CNNs) have been leveraged for intrusion detections due to its spatial feature extraction capability, although it fails to capture sequential data representations. Temporal CNN (TCN) [8] has been introduced by modifying the philosophy of CNN to include parallelism techniques and has shown great success over the CNN since TCN can learn long dependencies more convincingly. Besides, the structural design of the TCN is simpler and more precise compared to the RNNs, and TCN has achieved better performances than LSTM in many sequence problems [8].

A. Goals and Challenges

According to an exhaustive study of the recent works for IDS, there are a number of limitations and challenges that are addressed in this article.

1) *Spatial Relationships:* In IoT traffic communications, the samples recorded at neighboring positions are associated with one another creating a kind of residential interdependence.
2) *Sequential (Temporal) Relationships:* IoT traffic samples acquired at neighboring time periods are extremely closely related so the temporal context needs to be considered in designing new IDS.
3) *Data Heterogeneity:* IoT traffic streams are heterogeneous in nature either in the spatial domain or time domain so the participation of each kind of information is not always identical. As an example, due to the differences in IoT devices, routers, networking protocols, and manufacturers, the spatial information differs extensively, and the corresponding chronological periodicity also differs.
4) *Labeled Data Shortage:* Owing to the rapid and extreme increase in the size of IoT data, it is difficult to obtain up-to-date labeled IoT traffic data sets especially for intrusion/cyber-attacks detection. Thus, learning from unlabeled data has emerged as a vital research challenge in IoT environments.

B. Primary Contributions

This study presents the next innovative contributions to address the previously stated challenges.

1) A novel semi-supervised DL model for intrusion detection (SS-Deep-ID), which aims to detect intrusions/cyber-attacks in traffic records generated from IoT networks, making use of the benefits of both labeled and unlabeled traffic sequences during training.
2) A novel multiscale residual temporal convolutional (MS-Res) module is introduced to improve the network capability in learning spatiotemporal representation using residually connected dilated casual convolution.
3) A novel traffic attention (TA) module is introduced to calculate the weight representing the importance of different features, which helps the network emphasize the most significant feature for detecting intrusions.
4) An innovative semi-supervised hierarchical training approach is introduced for SS-Deep-ID in which we split the unlabeled IoT traffic records into numerous parts in consecutive order. Then, training procedures are performed on each separate part in a gradual strategy that enables preserving sequential interrelationships during training.

C. Paper Structure

The structure of this study is organized as follows. Section II describes the current studies relevant to intrusion detection in IoT traffic. Section III discusses the details of the proposed DL approach. Section IV presents the experimental configurations. Section V provides the results, analysis, and corresponding discussion. Section VI discusses the main limitations of this work and Section VII presents the study conclusions. Finally, future research directions are presented in Section VIII.

II. Related Work

Numerous research studies have investigated the realm of cyber-attacks on IoT communications [6], [39]. Artificial intelligence approaches have continuously been an important challenge for developing reliable IDS in an IoT-enabled environment [12], [15], [39]. In this regard, the current approaches for detecting IoT intrusions can be categorized into three distinct groups: 1) supervised approaches; 2) semi-supervised approaches; and 3) unsupervised approaches.

A. Supervised IDS

Supervised ML or DL approaches are typically trained using labeled IoT records to discriminate normal records from other attack records (i.e., binary classification) or to discriminate different attack classes from the normal traffic (i.e., multiclass classification). For example, Yang *et al.* [28] employed the k-nearest neighbors (kNNs) to develop a secure IDS for large-scale IoT data. Gao *et al.* [24] separately employed LSTM and feedforward neural network (FNN) for detecting intrusions and also experimented with them in an ensemble architecture. Similarly, Wu *et al.* [23] introduced an LSTM-Gauss-NBayes architecture that combined LSTM for processing temporal patterns in IoT data and the Gaussian Bayes algorithm for estimating the probability of having outliers. Zhou *et al.* [31] developed an IDS depending on reassembled feature representation using variational LSTM to alleviate the discrepancy between feature maintenance and dimensionality lessening. Moreover, in an attempt to improve the performance of supervised approaches, an

attention mechanism [37] has been recently employed after recurrent or convolutional layers to help the network focus important information. For example, Abdel-Basset *et al.* [19] proposed to detect intrusions using Deep-IFS that integrated GRU and multihead attention mechanisms (AMs). Liu *et al.* [14] introduced a federated learning approach to afford cooperative and decentralized training on edge devices and then exploited LSTM to capture temporal representations and attention augmented CNN learn important spatial information.

Despite the great achievement of supervised approaches in IDS, they did not gain much popularity because of the shortage in annotated IoT data, which requires exhaustive efforts and a long time [22]. Additionally, their performance is poor when the data is not evenly distributed between classes (class imbalance problem). This motivated us to use semi-supervised learning to develop efficient A-IDS.

B. Semi-Supervised IDSs

Semi-supervised approaches train a specific classifier using annotated and unannotated samples, particularly when a small volume of labeled samples is available. Numerous such techniques have been developed for detecting intrusions in IoT traffic and have demonstrated good performance [40], [41]. For instance, Ravi and Shalinie [29] developed a semi-supervised ML approach for detecting intrusions/cyber-attacks by integrating supervised neural networks and k-means augmented with repeated random sampling for unsupervised data clustering. Cheng *et al.* [22] introduced hierarchical stacking expansion for TCN that enable efficient detection of IoT anomalies, which reported 98.22% accuracy in detecting anomalies. Ravi and Shalinie [21] trained the extreme learning machine (ELM) algorithm in a semi-supervised manner to detect and mitigate DDoS attacks which demonstrated its effectiveness compared to other ML algorithms [40], [41]. Li *et al.* [17] introduced a collaborative IDS that incorporates a disagreement strategy in a semi-supervised training strategy, which achieved a hit rate of 94.23%. Gamage and Samarabandu [15] experimentally investigated the semi-supervised IDS using autoencoder (AE) and FNN; they also used deep belief networks (DBNs) in conjunction with FNN and evaluated these two approaches on four different data sets, and they realize much improved performance (accuracy: 98%–99%). Despite the success realized by these approaches, they still have lower performance in recognizing formerly identified attacks compared to supervised approaches. In other words, they take a longer time for training and testing to realize a robust performance [5]. Though this tradeoff, developing a semi-supervised IDS is still a very important and challenging task because of the large amount of unlabeled IoT data that cannot be exploited by supervised IDS as well as because of the heavy computations and untrustworthiness of unsupervised IDS [6]. This study, therefore, aims to address this tradeoff by developing an accurate DL model that can effectively learn the spatial-temporal representations from both labeled and unlabeled data in a reasonable amount of time.

C. Unsupervised IDSs

Intrusion detection approaches designed with no *clean* data or traffic labels are referred to as unsupervised IDS. These approaches are favorable as they do not require any traffic labels and are therefore cost efficient, and they utilize the inherent characteristics of IoT traffic samples to distinguish different attacks. Hence, they are reliably able to identify new attacks. As an example, Ergen and Kozat [30] exploited an LSTM architecture to process sequences of IoT traffic and to generate a fixed-length sequence. They then used the one-class SVM and support vector data description technique to calculate the final classification decision. Vu *et al.* [20] developed a regularized version of AE architecture to learn the latent representation of input traffics, which is exploited to fine-tune the performance of the supervised learner. Additionally, de Araujo-Filho *et al.* [18] introduced a fog-based modified generative adversarial network (GANs) called FID-GAN for detecting cyber-attacks with comparatively low latency as the computation become closer to user's devices. Their model estimates the reconstruction loss based on the reformation of data samples transformed into the latent space. Moreover, Wu *et al.* [11], modeled the anomaly detection in form of a one-class classification task where the training data contains the normal traffic data. They proposed a fault-attention probabilistic GAN that to automatedly find low-dimensional information implanted at the input with high-dimensional space, while taking the advantages of AE to reduce the loss of information that occurred during feature extraction. Li *et al.* [10] employed variational AE (VAE) as a network baseline which was then fulfilled by an RNN learn latent temporal representations from input time series. They also parameterized the average and variance for each time window using FNN to provide a nonstationary architecture that could operate with no persistent noise as usual. Despite the merits of unsupervised approaches, they still suffer from two primary issues. First, they do not have the robust performance of supervised approaches, particularly in recognizing formerly identified attacks. Second, they exhibit high computational complexity, which limits their applicability in real-time or resource-constrained IoT applications [7], [30], [40].

In summary, the present semi-supervised approaches show a great tradeoff between performance and computation. Thus, unlike recurrent networks, we propose to take the advantage of TCN to effectively capture spatial representations and sequential characteristics of IoT traffic data, which are essential details for realizing optimal performance. Moreover, we introduce an AM to help the network focus important features, hence mitigate the impact of noisy data and speed up network learning. Furthermore, the current semi-supervised approaches are also unable to maintain the sequential characteristics of IoT data during training [22]. Thus, we propose to train the lightweight SS-Deep-ID hierarchically.

III. Methodology

This section presents a detailed discussion of the proposed semi-supervised DL for detecting intrusions (called SS-Deep-ID) in IoT traffic. Fig. 1 presents the main

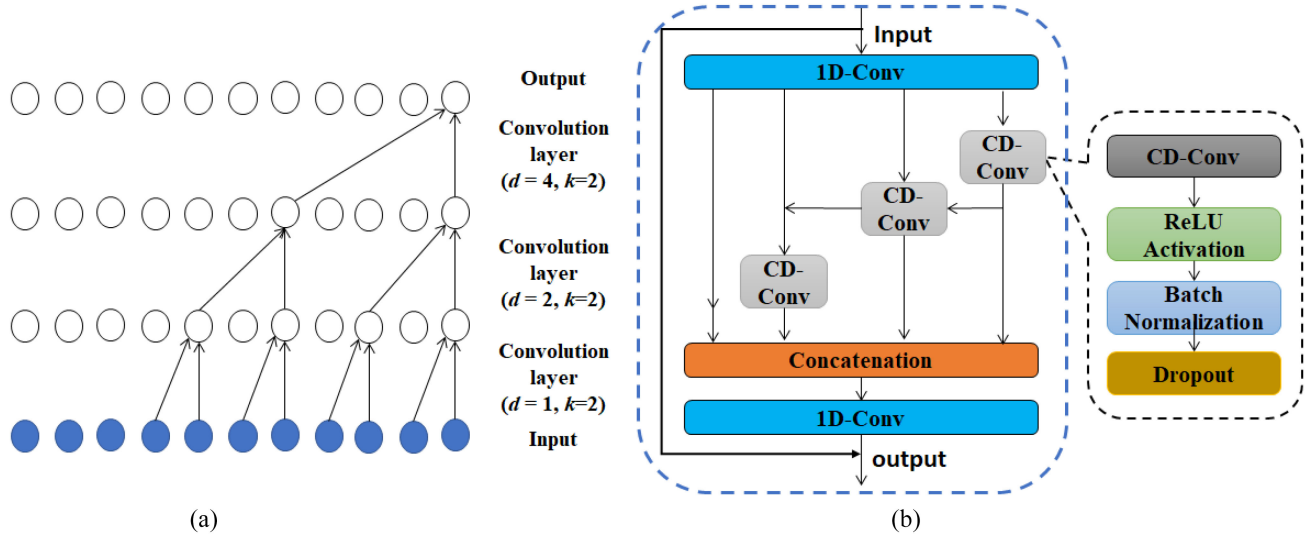

Fig. 1. (a) Representation of casual convolution. (b) Architecture of proposed MS-Res.

components of SS-Deep-ID, and Fig. 2 displays the final architecture which comprises three primary modules. First, the MS-Res block was proposed to finetune the network capabilities in capturing the spatial-temporal representation. Second, an attention module is introduced in parallel with the MS-Res block to quantify the importance of the input representation to help the network focus on representative patterns. Finally, the output of these modules is concatenated and fed into the fully connected layer to calculate the probability that the underlying input belongs to a specific class.

A. Multiscale Temporal Convolution

Recurrent models, such as RNN and LSTM are dedicated to processing sequential data by using applying a gating mechanism to act as a memory for preserving information from the previous history. In comparison, CNNs are often used for feature extraction in static contexts, but CNN models are inappropriate for modeling sequential data due to the lack of a memorization mechanism for a long-term dependency [28]. Nevertheless, later studies demonstrated the efficiency of new types of CNN models (i.e., TCN), that could perform better than RNN in the area of language translation and video and audio synthesis [9], [30]. Motivated by the success of TCN in a variety of applications, we propose to redesign and improve the TCN architecture for intrusion detection and attack classification from IoT traffic data.

The key constituent of the TCN is dilated causal convolutions (DC-Conv) in which causal convolution (C-Conv) layers [29] permit certain production at time t to be just influenced by the input at present time t as well as the earlier input (i.e., $t-1,\ t-2,\ t-3,\ \ldots$). Furthermore, different receptive fields are employed to empower the CNN capability to handle the records in the input steams [31]. A wider receptive field designates a wide ranging of traffic at the input layer. Fig. 1(a) presents the receptive field of a typical C-Conv. The output (i.e., $\hat{y}_t$) rests on the input samples from the present time step (i.e., x_t) and the former input time step (e.g., x_{t-1}, x_{t-2}, and x_{t-3}). The convolution procedure F for C-Conv stack at the time step t is articulated in equation

$$F(t) = \sum_{i=0}^{k-1} f(i) X_{t-i} \tag{1}$$

where $f\ (i\ )$ represents the ith kernel of the corresponding layer [i.e., the hidden layer in Fig. 1(a)], k denotes the kernel size and $t-i$ represents the input at the previous time step.

As for the typical C-Conv stack with the kernel size $k = 2$ and depth $d = 4$ in Fig. 1(a), only four inputs impact the output owing to the restricted receptive field.

This discloses a key drawback of the typical C-Convs layers, which is a very large kernel size k or a very deep model is required to preserve a great receptive field, that assists making the model production to be affected by long and effective ancient input records. Nevertheless, a larger value of k might prevent network convergence [30], which will degrade the classification accuracy. Further increasing the network depth may stabilize the training and result in significant performance degradation [7].

To tackle the former problem, TCN exploits the DC-Conv, where the rate of dilation d rises exponentially when the depth of the network increases. In this manner, TCN might not only successfully enlarge the receptive field with no need for enlarging the value of k, but also have satisfactory computational performance. As an example, the outcome $\hat{y}_t$ hinges on all incoming traffic samples with limited processing operations. Hence, the convolution procedure F for the DC-Conv layers in the time step t is formulated as shown in

$$F(t) = \sum_{i=0}^{k-1} f(i) \cdot X_{t-d \cdot i}. \tag{2}$$

Having an adequate kernel size, the steadiness of training deeper TCN turns out to be a major consideration for realizing a required classification performance. To tackle this limitation, motivated by [33], a hierarchical multiscale residually connected structure is introduced to form the MS-Res module

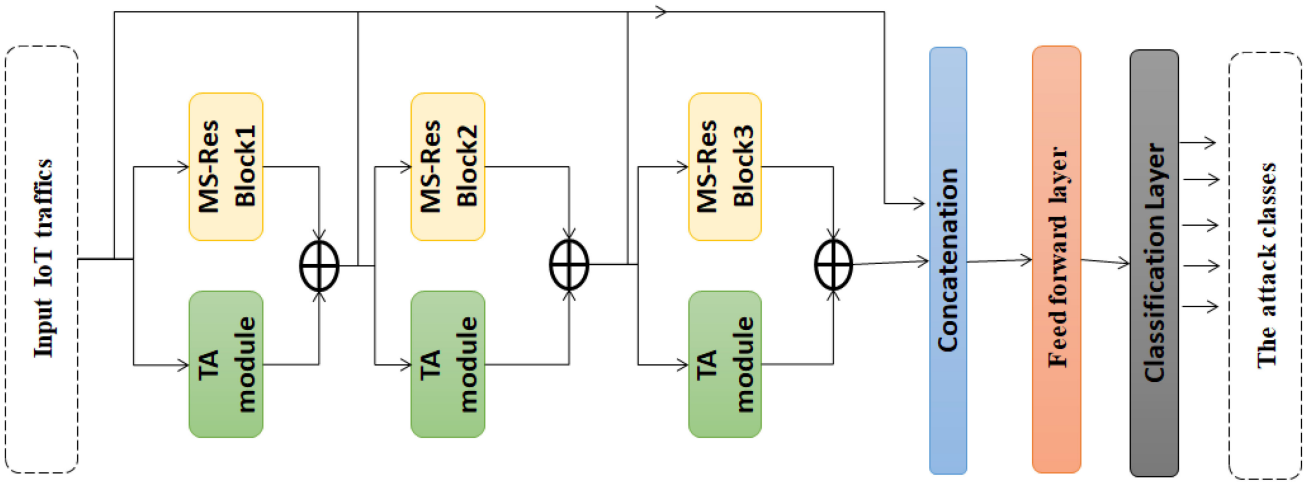

Fig. 2. Architecture of the proposed SS-Deep-ID.

presented in Fig. 1(b), comprising a variety of layers. In particular, the input is passed into 1D-Conv to reduce the input dimension and to lessen the computational workload. Then, the convolutional output is passed to the subsequent four parallel paths. The leftmost path is a simple skip connection offer a kind of feature reusability through training; other paths are established utilizing DC-Conv layers. The DC-Conv layer consists of a DC-Conv activated using rectified linear unit (ReLU) as an activation function, batch normalization (BN) [42] layer, and dropout [43] layer in sequential order. The main purpose of the BN layer is to help to stabilize network performance during training hence speed up the model convergence. The dropout layer is employed to avoid overfitting problems during training.

To assure that the proposed SS-Deep-ID is computationally lightweight, DC-Conv layers were employed to replace traditional convolutions. The later 1D-Conv filter acquires features from the received input maps f_i and follows the wide range of receptive fields to process the acquired information. In particular, the production of a certain path is passed to the concatenation layer and simultaneously used as an input to the succeeding path via an elementwise addition. This process is duplicated multiple times until the feature maps generated from all paths are extracted. This operation could be formulated as in

$$F_i = \begin{cases} f_i, & i = 1 \\ \text{DC-Conv}(f_i), & i = 2 \\ \text{DC-Conv}(f_i + F_{i-1}), & 2 < i \leq 4 \end{cases} \tag{3}$$

where f_i feature map at path i and the F_i characterize the calculated output map from each path.

B. Traffic Attention Mechanism

The AM was introduced in [37] to increase the network emphasis on the significant characteristics. According to this, AM has been widely used to expand the feature extraction capability of DL models in several domains, i.e., image and language processing [37]–[39]. Commonly, heterogeneous IoT traffic data always contain different features that they vary in the degree of importance for detecting intrusions. Thus, inspired by [37], we propose a TA module to empower the model performance by focusing on the important features in the received IoT traffic data.

Assuming the input traffic is set to be $I^x = \{I_1^x, I_2^x, \ldots, I_n^x\}$, The TA module exploits the incoming traffic data and computes the significance score according to the equations

$$\sigma_j = \frac{\exp\left(I_j^x.w\right)}{\int \exp\left(I_j^x.w\right)} \tag{4}$$

$$c = \sum_j \sigma_j.I^x{}_j \tag{5}$$

where w represents the arbitrarily initialized weight vector that is optimized during the training, σ_j represents the normalized score of importance, and c denotes the context feature of the present timestamp computed according to the score σ_j.

After that, an elementwise multiplication is performed on the output of the MS-Res module and the output of the TA module. Assuming that input $I^x = \{I_1^x, I_2^x, \ldots, I_n^x\}$ $(1 < x < X)$, the output of the Ms-Res and TA modules are denoted as $W_{\text{MS-Res}}(x, c)$ and $W_{\text{TA}}(x, c)$, respectively. Then their multiplication is represented in equation

$$W(x, c) = W_{\text{MS-Res}}(x, c) \odot W_{\text{TA}}(x, c) \tag{6}$$

where x denotes the relative position of the corresponding position at the input vector, $\odot$ represents the elementwise multiplication operator, and c represents the channel.

Succinctly, the TA module is proposed to enable the model to acquire a more inclusive contextual representation, in that way capture the significant IoT features of the received traffic sequence. This is performed by suppressing the interfering of insignificant features to the model, hence helping the model to discriminate the importance of various IoT features.

C. Classification Layer

In this section, the outputs of MS-Res blocks and TA blocks are exploited to calculate the final traffic class. A feed-forward (FF) layer is encoding the captured spatiotemporal

representations into a linear representation appropriate for predicting the final class label using SoftMax operation where each class is assigned a certain probability score and the class with the highest probability is regarded as the final model prediction as formulated in equations

$$\text{pr} = \text{SoftMax}(X) = \frac{\exp(X)}{\int_1^c \exp(X)} \tag{7}$$

$$\tilde{y} = \text{argmax}(\text{pr}) \tag{8}$$

where X denotes the output of the FF layer, and PR represents the probability score. Training the model to require minimization of cross-entropy loss was calculated with equation (9)

$$\text{Loss} = -\sum (y_i \cdot \log \tilde{y}_i + (1 - y_i)\log(1 - \tilde{y}_i)) \tag{9}$$

where y_i is the actual label and $\tilde{y}_\text{i}$ is the model predicted label.

D. Semi-Supervised Training

Owing to the huge amount of IoT traffic records generated daily, and the effort and time required to label every record, semi-supervised learning has become favored to train a DL model using a combination of a huge number of unannotated records and a small number of labeled records. Hence, the semi-supervised model is more suitable for anomaly detection in IoT communication. This study set 75% of training data is unlabeled, and the remaining was used as labeled samples. The labeled traffic is represented by $\{X_l, Y_l\}$ where $X_l = \{x_{l1}, x_{l2}, \ldots, x_{ln}\}$ and the corresponding labels $Y_l = \{y_{l1}, y_{l2}, \ldots, y_{ln}\}$. The unlabeled records are represented by $X_u = \{x_{u1}, x_{u2}, \ldots, x_{um}\}$ with no labels. Generally, the research community recommends labeling the IoT traffic records that are previously acquired. This indicates that the timestamp of labeled records is preceding comparative to the unlabeled records. Moreover, it is well-thought-out that these IoT traffic records are sequential as they are aggregated over an elongated period. In particular, the time duration of unlabeled data is longer than the labeled data. If there is no operation for an unlabeled data set, it will neglect the sequential relationship when training unlabeled IoT records. Thus, we employ a hierarchical method to split the unlabeled samples into comparatively short durations.

To train the proposed SS-Deep-ID hierarchically, we split the unlabeled part of the data into P parts. The split data is referred to as $X_u = \sum_{i=1}^{P} X_u^i$. Significantly, part $P-1$ precedes part P. Increasing the value of P improves the network's representation power (spatiotemporal) as it will enable improving the network learning capability every time a new part is considered. However, such an increase in the value of P might result in overfitting, and raise the computational cost [22]. Thus, investigating the optimal value of P is vitally important. Then, the SS-Deep-ID can be evaluated with each unlabeled part X_u^i, using the labeled data and the previously experimented unlabeled data $\sum_{j=1}^{i-1} \grave{X}_u^j$. Each experiment, once complete, results in a data set called $\grave{X}_u^j$. Accordingly, the final set attained from all parts is denoted as $\sum_{i=1}^{P} \grave{X}_u^i = \grave{X}_u$. Hence, the experiment of SS-Deep-ID on every part of the unlabeled traffic records is formulated in equation

$$\grave{X}_u^i = \text{SS-Deep-ID}_i\left(X_l + \sum_{j=1}^{i} \grave{X}_u^j\right). \tag{10}$$

After processing all unlabeled parts when all layers of HS-TCN complete the training process, the unlabeled data are all assessed. The final decision output is computed using both labeled and experimented data to determine the intrusions/cyber-attack class according to equation

$$\text{F} = \text{SS-Deep-ID}\left(X_l + \grave{X}_u\right). \tag{11}$$

According to (11), the hierarchical schema enables splitting the unlabeled records into multiple parts with restricted time duration. Throughout the experiments of unlabeled traffic data, the ith layer of the SS-Deep-ID experiments with unlabeled traffic X_u^i by incorporating the output of the preceding experiments $\grave{X}_u^{i-1}$ which indicates that the experiments are performed gradually. Accordingly, this semi-supervised hierarchical training takes into account the chronological interdependence in the unlabeled IoT records in an incremental strategy and is ideally suited to the traffic data in IoT networks.

E. Deployment in IoT Environment

This section deliberates the way the proposed SS-Deep-ID is deployed and operated in real-world IoT networks. In view of this, Fig. 3 present a system diagram for the semi-supervised intrusion detection framework in fog-enabled IoT networks using the proposed SS-Deep-ID. It could be noted that the structure of SS-IDF primarily consists of three layers, namely, cloud layer, fog layer, and edge layer. The cloud layer is well-known for its high and powerful computational resource; hence, the process of model training is performed on this layer as the training requires access to a massive amount of IoT traffic data. Such big data could be easily aggregated and stored in the cloud. Moreover, the cloud layer also stores the model configurations, older pretrained versions, and other settings related to training transactions.

The fog layer typically consists of many fog servers/devices to bring the computation closer to the edges of the IoT network. In the SS-IDF, the fog layer has a critical role as it is where the intrusion detection would happen. Specifically, every fog node consists of four main components, i.e.: 1) traffic aggregation component; 2) traffic preparation component; 3) traffic countermeasures components; and 4) traffic diagnosis. The traffic aggregation component is accountable for capturing and receiving the IoT traffic records from the connected part of the edge IoT network, then pass the batched samples for the preparation phase. The traffic preparation component is responsible for transforming the received batches into a standard format, applying necessary data cleaning, and normalization. After that, the traffic diagnosis component is designated using the trained SS-Deep-ID to classify the prepared IoT traffic data without any communication with the cloud backend, hence preventing any delay. This classification process can be carried out in binary class or multiclass scenario. once an activity is identified as an attack, the

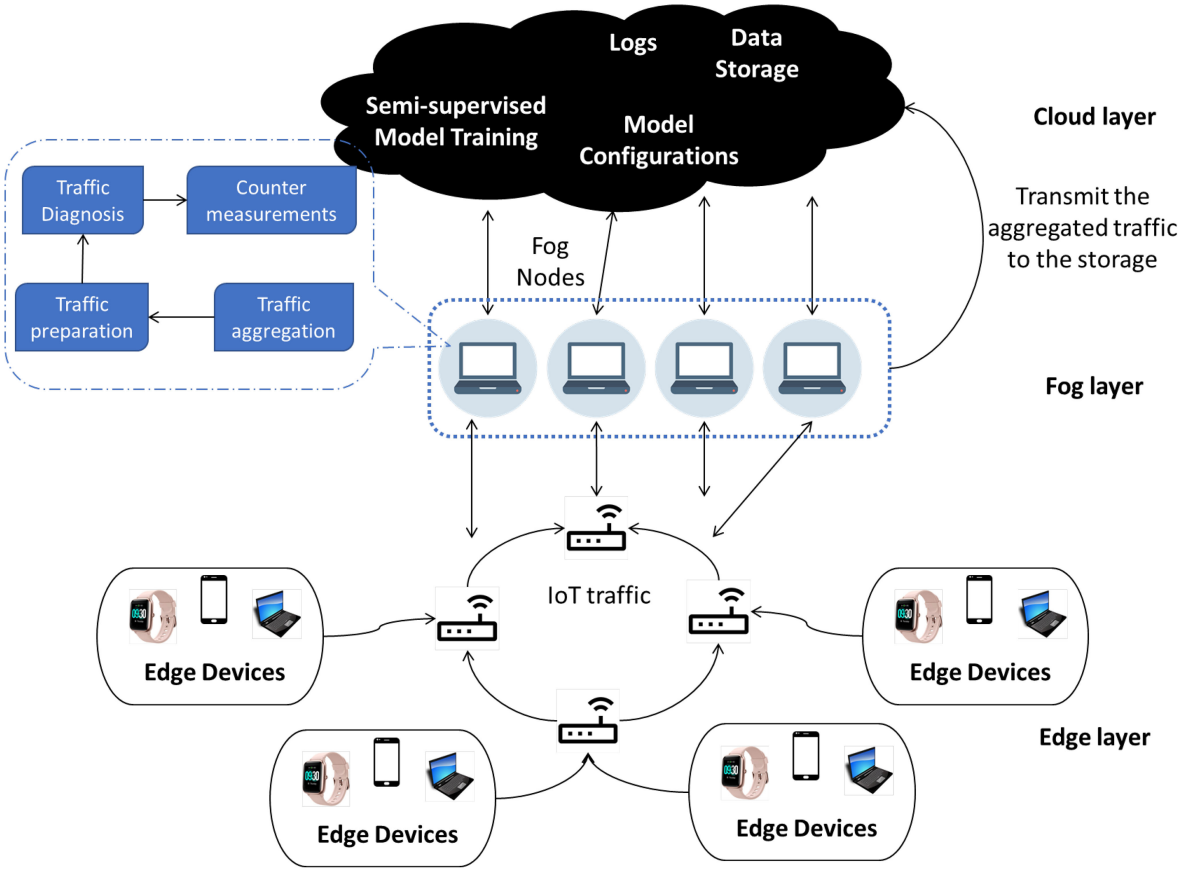

Fig. 3. Systematic diagram indicating of semi-supervised intrusion detection framework in fog-enabled IoT networks.

information from the provided activity is forwarded to the cloud backend for brief. Every fog node is accountable for diagnosing the associated zone of the IoT network. Thus, all IoT traffic records are caught by the corresponding fog device operating in promiscuous mode. For example, given substantial alterations in the traffic on a specific zone of the underlying IoT network, if these alterations are malevolent, i.e., denial-of-service (DoS) incidents, the proposed SS-Deep-ID would recognize them and communicate the countermeasure components. When the alteration is benign, it will be essential to connect some nodes to an alternative accessible fog device to mitigate it from turn out to be congested. After all, the countermeasures component takes the decision generated from the SS-Deep-ID and then performs the necessary predefined countermeasure module, i.e., warnings, blocking, and removing actions. Follow this, information about the identified action is transmitted to the cloud for the results' log component.

Finally, the edge layer consists of the edge node and edge devices (i.e., laptops, smartphones, smartwatches, etc.) communicating through IoT networks via routing and switching devices and simultaneously connected with a specific fog server/node as a computation bridge to the cloud backend.

IV. Experiments

A. Data Set Description

A wide range of IDS data sets are accessible to the public; among them, we choose two recent data sets for evaluating the performance of the proposed model, namely, the CIC-IDS2017 [31] and the CIC-IDS2018 [32]. The CIC-IDS2017[1] consists of 2 830 743 IoT traffic samples acquired in a tiny, simulated network environment in which there are six types of recent attacks initiated from a dispersed network. The data is acquired by gathering measurement of eight traffic monitoring sessions, then stored in eight comma-separated value (CSV) files. The data comprise 78 regular features and one class label that includes benign traffic and other 14 categories of attacks. On the other hand, the CIC-IDS2018[2] is broadly known real-world heterogeneous intrusion detection data that are typically more complex and include missing values, irrelevant features, outliers, erroneous instances, and high discrepancies. Thus, more similar to the real-world IoT networks. It comprises 16 233 002 IoT traffic samples gathered from ten days of network traffic. Nevertheless, the samples are captured on a wider network denoted as victim networks with 30 servers and 420 client machines. The data set consists of 79 attributes and has around 17% of the samples as attack traffics. It is distributed over ten CSV files that are publicly available for download. The details of the two data sets and their corresponding data distribution employed in our experiments are shown in Table I.

[1]https://www.unb.ca/cic/datasets/ids-2017.html
[2]https://www.unb.ca/cic/datasets/ids-2018.html

TABLE I
CHARACTERISTICS OF THE DATA SETS USED TO EVALUATE THE PROPOSED SS-DEEP-ID

Dataset	No. original samples	No. of features	No. of classes	Attacks	Data Distribution		
					Classes	Train	Test
CIC-IDS2017 [31]	Original: 2,830,743 Removed: 2867 Cleaned: 2,827,876	categorical:1, numeric:77	15 classes mapped to seven classes	Bot, DDoS, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, FTP-Patator, PortScan, SSH-Patator, Heartbleed, Web Attack Brute Force, Web Attack XSS, Web Attack SQL Injection	Benign	1816543	454135
					Bot	1544	385
					Dos	304419	76104
					Infiltration	29	7
					PortScan	127024	31756
					Web Attack	1733	433
					Brute Force	11012	2752
CIC-IDS2018 [32]	Original: 16,233,002 Removed: 782,296 Cleaned: 15,450,706	categorical:1, numeric:79	15 classes mapped to seven classes	SSH-Brute force, FTP-Brute Force, Brute -Force -XSS,Brute -Force -Web SQL-Injections, DoS-attacks-Hulk, DoS-attacks-SlowHTTPTest, DoS-attacks Slowloris, DDoS attacks GoldenEye, DDOS-attack-HOIC, DDOS-attack-LOIC-UDP, DDOS-attack-LOIC HTTP, Bot, Infiltration.	Classes	Train	Test
					Benign	10158176	2539543
					Bot	228953	57238
					DoS	494036	123508
					Infiltration	129548	32386
					DDoS	1044353	261088
					Web attack	154758	38689
					Brute Force	150744	37686

B. Data Preparation

The next preparation procedures were conducted on both data sets before training. First, we eliminated redundant features (i.e., Fwd_Header_Length) and illegal IoT flow records, i.e., null or missed values, or a character in arithmetical fields. Hence, 2867 records were eliminated from the CIC-IDS2017 data, and 782 296 records were eliminated from the CIC-IDS2018 data set. Second, the operation of one-hot encoding is applied to categorical features to convert them into a numerical representation. Third, as defined in [15], [31], and [32], the two data sets show high-class discrepancy which occasionally results in a high false alarm. Inspired by the notion presented in [44]–[46], the original classes are mapped into new traffic labeling that combines similar classes as shown in Table I. Fourth, the data were normalized by scaling all features in a standard manner. Fourth, each data set was divided into three main groups, and the division was performed in the stratified mode for keeping the original distribution of samples across different classes. As a result, 60% of data was used for training data, 20% of data was used for validation, and 20% was employed for testing.

C. Evaluation Measures

To assess the performance of the proposed model, the following evaluation metrics are employed in this study.

Accuracy: Represent the average correctly identified IoT traffics by the proposed model and is computed according to the equation

$$\text{Accuracy (ACC)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100. \quad (12)$$

Precision: A measure of proportion of positively identified samples that was essentially positive and is defined in equation

$$\text{Precision (PRC)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100. \quad (13)$$

Recall: A measure of the proportion of real positive instances that got detected as positive and is defined according to the equation

$$\text{Recall (RCL)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100. \quad (14)$$

F1-Measure: A measure of the weighted average of the PRC and RCL, it must lie within the interval [0, 1] where it reaches its finest value at 1 and most awful at 0 and is calculated according to the equation

$$\text{F1-measure (F1)} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (15)$$

where false negative (FN), also called the Type II error, represents the number of positive traffic samples mistakenly recognized as negative. The false positive (FP), also called Type I error, denotes the number of negative traffic samples mistakenly recognized as positive. The true positive (TP) represents the number of positive traffic samples rightly recognized as positive. The true negative (TN) is the number of negative traffic samples rightly detected as negative.

D. Implementation Setup

The proposed SS-Deep-ID was implemented using the Python 3.7 programing language, utilizing Keras API deployed on top of the TensorFlow backend; the training and inference were performed on a PC equipped with a 64-bit version of Windows 10 on an Intel Xeon E5-2670 CPU@ 2.60 GHz, with 256-GB RAM, and graphics card

TABLE II
CONFUSION MATRIX OF THE PROPOSED SS-DEEP-ID ON THE CIC-IDS2017 DATA SET IN CASE OF BINARY CLASS SCENARIO

		Predicted values			
		Benign	Attack	RCL (%)	F1 (%)
Actual values	Benign	453346	789	99.82	99.74
	Attack	1517	109923	98.63	98.96
	PRC (%)	99.66	99.28		

TABLE III
CONFUSION MATRIX OF THE PROPOSED SS-DEEP-ID ON THE CIC-IDS2018 DATA SET IN CASE OF BINARY CLASS SCENARIO

		Predicted values			
		Benign	Attack	RCL (%)	F1 (%)
Actual values	Benign	2528231	11312	99.55	99.59
	Attack	9467	541128	98.28	98.12
	PRC (%)	99.76	98.86	-	-

(NVIDIA QUADRO P4000) 8-GB memory. Several experiments are performed to determine the optimal parameters of SS-Deep-ID. The model is initialized using the uniform distribution and is optimized using a well-known Adam optimizer with an initial learning rate of 0.0003. The dropout rates were initialized to be 0.25, a batch size of 256 was employed to experiment with the model.

V. DISCUSSION AND ANALYSIS

A. Results

This section discusses the result attained by the proposed SS-Deep-ID in two scenarios, namely, binary-class scenario and multiclass scenario. Particularly, in the binary classification, the IoT traffic is classified as either benign or attack, while the latter scenario employs the SS-Deep-ID to classifies the IoT traffic samples into seven classes previously mentioned in Table I.

For the binary class scenario, the confusion matrix of the SS-Deep-ID on the CIC-IDS2017 and CIC-IDS2018 data is presented in Tables II and III, respectively. It could be observed that although of the large difference between the number of benign samples and the number of attack samples, the SS-Deep-ID achieve robust performance with accuracy above 99% and F1-measure between 98% and 99%.

For the multiclass scenario, the confusion matrix of SS-Deep-ID on the CIC-IDS2017 is tabulated in Table IV. It could be seen that the infiltration class got the lowest performance (precision of 75% and F1-measure of 80%), followed by the Web attack class (precision of 85.65% and F1-measure of 91.37%), then the bot class (precision 91.54% and F1-measure of 93.51%). This could be justified as a side effect of class imbalance in the data as the number of samples in these three classes is much smaller than the other. The other classes of IoT traffic realized an efficient detection performance with an F1-measure ranging from 96.59% up to 99.85%. Besides, the confusion matrix of SS-Deep-ID on the CIC-IDS2018 is shown in Table V. It is observable that the Brute force traffic got the lowest performance (precision of 91.72% and F1-measure of 91.75%), followed by the Web attack traffics (precision of 93.07% and F1-measure of 92.09%), followed by the infiltration class (precision 93.64% and F1-measure of 93.37%). Despite the high discrepancies in the number of samples of these categories, the proposed SS-Deep-ID still showing good performance. It is also visible that there is comparatively high confusion between DoS traffic and DDoS traffic resulting in an F1-measure of 93.64% and 96.79%, respectively. compared with CIC-IDS2017, the incorrect recognition of benign activities is comparatively high because of the noisy and heterogeneous nature of this data set.

B. Comparative Analysis

This section compares the performance of the proposed SS-Deep-ID against recent state-of-the-art methods (i.e., supervised and semi-supervised) on the two data sets as shown in Table VI and VII. Where the former provides the results of competing approaches in the binary class scenario, and the latter tabulates the comparative results of the multiclass scenario. The motive behind selecting the semi-supervised competitive models is that their performance is as robust as the supervised models, as later discussed. The previous deep networks used in comparisons achieved similar results to the ones stated in the corresponding studies. Consequently, these experiments confirm the rightness of our implementation and demonstrate that the ML algorithms are properly trained.

In the binary class scenario, the comparative results are presented in Table II, and it shows that the supervised SVM and RF show good performance on both data sets, and they are appropriate ML techniques to take into account when developing new ML approaches when designing the classification layer as later discussed. Besides, the LSTM [47] is included in these comparative experiments owing to its ability to learn the temporal representations in IoT traffics, which might enhance the classification performance of IDS. However, the observed results of LSTM show that its performance (accuracy: 98.61% and F1-measure: 98.67%) is still less than the other approaches. Specifically, it shows a high false-negative ratio in the case of time-associated classes, i.e., Brute force and DoS. Furthermore, we observed that increasing the sequence length has no impact on the performance which may be because that input of window of interrelated records captures the important time correlations at the record level through the input features. This interrelationship might encompass invaluable information to be learned by the LSTM. On the other hand, for the semi-supervised approaches, it is worth observing that on both data sets the performance of TSVM [36] has significantly degraded 83.25% of accuracy and 87.78% of F1-measure. This happens because of the lack for the ability of modeling complex spatiotemporal characteristic and mainly seeks to partition hyperplane by assigning a label to unlabeled traffic. Moreover, we could note that the performance of the AE+ANN [15] and DBN+ANN [15] realize

TABLE IV
CONFUSION MATRIX OF THE PROPOSED SS-DEEP-ID ON THE CIC-IDS2017 DATA SET IN CASE OF MULTICLASS SCENARIO

		Predicted values								
		Benign	Bot	DoS	Infiltration	PortScan	Web Attack	Brute Force	RCL (%)	F1 (%)
Actual values	Benign	453551	11	214	1	298	17	43	99.87	99.85
	Bot	2	368	1	0	7	3	4	95.58	93.51
	DoS	395	9	75585	0	63	19	33	99.31	99.39
	Infiltration	0	0	0	6	1	0	0	85.71	80.0
	PortScan	309	7	168	1	31211	25	35	98.28	98.54
	Web Attack	3	3	1	0	2	424	0	97.92	91.37
	Brute Force	49	4	10	0	4	7	2678	97.31	96.59
	PRC (%)	99.83	91.54	99.48	75	98.81	85.65	95.88	-	-

TABLE V
CONFUSION MATRIX OF THE PROPOSED SS-DEEP-ID ON THE CIC-IDS2018 DATA SET IN CASE OF MULTICLASS SCENARIO

		Predicted values								
		Benign	Bot	DoS	Infiltration	DDoS	Web Attack	Brute Force	RCL (%)	F1 (%)
Actual values	Benign	2527941	1298	2757	918	2911	1811	1907	99.54	99.51
	Bot	2479	53764	182	198	343	198	74	93.93	95.07
	DoS	2164	198	115566	180	5037	185	178	93.57	93.64
	Infiltration	1612	89	108	30154	172	47	204	93.11	93.37
	DDoS	2957	229	4213	167	253282	113	127	97.01	96.79
	Web Attack	1801	157	307	254	278	35258	634	91.13	92.09
	Brute Force	1949	134	194	334	214	273	34588	91.77	91.75
	PRC (%)	99.49	96.23	93.71	93.64	96.59	93.07	91.72	-	-

TABLE VI
RESULT OF COMPARATIVE EXPERIMENTS IN BINARY-CLASS SCENARIO

	CIC-IDS2017				CIC-IDS2018			
Study	**ACC (%)**	**PRC (%)**	**RCL (%)**	**F1 (%)**	**ACC (%)**	**PRC (%)**	**RCL (%)**	**F1 (%)**
Supervised Approaches								
SVM [35]	97.89	98.87	98.01	98.43	97.02	97.05	97.32	97.18
RF [44]	99.86	99.86	99.86	99.86	98.34	97.80	98.34	98.06
ANN [15]	99.58	99.56	99.58	99.56	98.38	98.54	98.36	98.44
LSTM [47]	98.61	98.02	99.33	98.67	98.01	97.54	96.5	97.01
Semi-Supervised Approaches								
TSVM [36]	83.25	84.15	91.75	87.78	79.31	80.12	89.87	84.71
AE + ANN [15]	98.12	98.1	98.12	98.10	98.22	97.5	98.22	97.85
DBN+ANN [15]	98.97	98.07	98.42	98.24	98.31	97.61	98.31	97.95
SS-Deep-ID	**99.6**	**99.48**	**99.23**	**99.35**	**99.33**	**98.78**	**98.91**	**98.85**

TABLE VII
RESULT OF COMPARATIVE EXPERIMENTS ON DATA SETS

	CIC-IDS2017				CIC-IDS2018			
Study	**ACC (%)**	**PRC (%)**	**RCL (%)**	**F1 (%)**	**ACC (%)**	**PRC (%)**	**RCL (%)**	**F1(%)**
Supervised Approaches								
SVM [35]	96.74	82.27	82.74	82.50	95.28	79.76	87.17	83.30
RF [44]	96.38	80.16	80.28	80.22	95.07	83.81	86.21	84.99
ANN [15]	95.38	81.34	80.91	81.12	95.97	82.34	84.94	83.62
LSTM [47]	97.84	87.42	85.82	86.61	97.01	87.07	92.67	89.78
Semi-Supervised Approaches								
TSVM [36]	89.11	75.14	72.18	73.63	79.31	80.12	87.87	83.82
AE + ANN [15]	98.09	90.11	87.64	88.86	95.98	91.97	90.73	91.35
DBN+ANN [15]	98.35	89.87	90.84	90.35	96.08	92.46	92.48	92.47
SS-Deep-ID	**99.69**	**92.31**	**96.29**	**94.18**	**98.71**	**94.91**	**94.30**	**94.92**

comparable performance across different evaluation measures (i.e., F1-measure: 97%–98%). Similarly, the proposed SS-Deep-ID shows that the performance is robust as with supervised approaches. This explains the usefulness of incorporating the unlabeled IoT data during training. Compared with semi-supervised approaches [15], [36], the proposed model realized around 1% improvements in accuracy and F1-measure on the CIC-IDS2017 and CIC-IDS2018 data set.

In the multiclass scenario, the comparative results are shown in Table II. In the scope of supervised approaches, it could be noted that SVM [35], RF [44], and ANN [15] attain the lowest performance with 82.50%, 80.22%, and 81.12%

TABLE VIII
RESULTS OF PAIRS-TEST ON DIFFERENT DATA SETS USING ACCURACY AND F1-SCORE MEASURES

	CIC-IDS2017		CIC-IDS2018	
Study	**ACC**	**F1**	**ACC**	**F1**
TSVM [36]	0.0014	0.0091	0.0031	0.0023
AE + ANN [15]	0.0211	0.0287	0.0211	0.0151
DBN+ANN [15]	0.0195	0.0234	0.0312	0.0214

of F1-measures, respectively, on CIC-IDS2017, meanwhile realizing 83.30%, 84.99%, and 83.62% of F1-measure on CIC-IDS2018. This could be explained by the absence of discrimination power between different attack categories, which could be handled by employing a robust feature engineering technique before the model. This shortcoming tried to be tackled using LSTM [47] which improved the F1-measure by 3% and 5% on CIC-IDS2017 and CIC-IDS2018 correspondingly. In opposite, concerning the semi-supervised approach, the TSVM shows similar behavior as in binary scenario. Moreover, AE+ ANN [15] and DBN+ANN [15] show relatively good performance with 88.86% and 90.35% of F1-measure, respectively, on the CIC-IDS2017. It also achieved 91.35% and 92.47% on CIC-IDS2018. This explains the effectiveness of the unsupervised pretraining in improving the performance of ANN where the DBN or AE is employed for efficient extraction and dimensionality reduction. Furthermore, it could be seen that the proposed SS-Deep-ID show obvious performance enhancements (accuracy: 1.34% and F1-measure: 3.83%) on CIC-IDS2017 and realized great performance on enhancement (accuracy: 2.63 and F1-measure: 2.45%) CIC-IDS2018 data set. This validates the effectiveness of our architecture and the efficiency of the introduced semi-supervised hierarchical training.

C. Statistical Significance

In order to determine the statistical significance corresponding to the discrepancies in the evaluation measures, a paired student's *t*-test is performed on the accuracy measure on both data sets for semi-supervised approaches, and the corresponding *P*-values are shown in Table VIII. The reported *P*-values is using SciPy library[3] [38] which is an open source scientific tools for Python. The results show statistical significance when the *P*-value < 0.05, otherwise they indicate no statistical difference between models' outcomes. It can be observed that the competing methods attain *P*-values lower than 0.05 which in turn gives clear evidence that the proposed SS-Deep-ID outperforms other competing approaches with statistical significance on the accuracy measure. For extra validation, a paired t-test was performed to compare the significance of results on the F1-measure on the two data sets. The resultant *P*-values indicate the statistical significance of the proposed model against the competing methods.

[3]https://scipy.org/

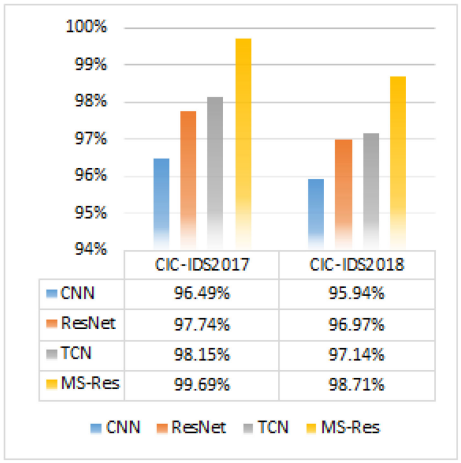

Fig. 4. Impact of TA module on the model's accuracy.

D. Ablation Studies

In this section, several ablation experiments are performed to help analyzing the model behavior by highlighting the contribution of different components on the final classification performance in the multiclass scenario.

1) Impact of MS-Res Module: This experiment was carried out to investigate the effectiveness of the proposed MS-Res module. In particular, three variants of the proposed SS-Deep-ID are redesigned by replacing the MS-Res module with two convolutional layers (i.e., CNN), residual block [48] and TCN module [22]. Then, we evaluate and compare this variant on the two data sets and report the corresponding accuracies and F1-measures in Figs. 4 and 5, respectively. It could be noted that employing the CNN result in the lowest performance with an accuracy of 94.49% and 94.94% and by achieving an F1-measure of 89.12% and 90.03% on CIC-IDS2017 and CIC-IDS2017 correspondingly. This is explained by the fact that the convolution layer fails to capture the temporal dependencies inherent in IoT traffic, also because of the information loss in later layers. To tackle the information loss between layers, the residual connection between layers is employed in [48] and improve the performance of this shown to increase the performance of SS-Deep-ID with 1%–2% improvements on accuracy and F1-measure, respectively. Moreover, the TCN-based architecture shows obvious improvements (i.e., 1%–2% increase) over the residual module [48], which could be justified by the abilities of temporal convolutions on capturing the chronological characteristics of IoT traffics. More intrinsically, the proposed MS-Res module result in 2%–3% accuracy improvements and 1.5%–3% improvements on F1-measure. This validates our hypothesis that multiscale cumulative processing of IoT traffic empowers the representation power of the network, hence finetune intrusion detection performance even in case of complex data.

2) Impact of the TA Module: This experiment was carried out to explore the effectiveness of the proposed TA module. We compare the performance of the proposed SS-Deep-ID with and without TA modules on both data set and report the achieved performance in Fig. 6. it could be seen that the

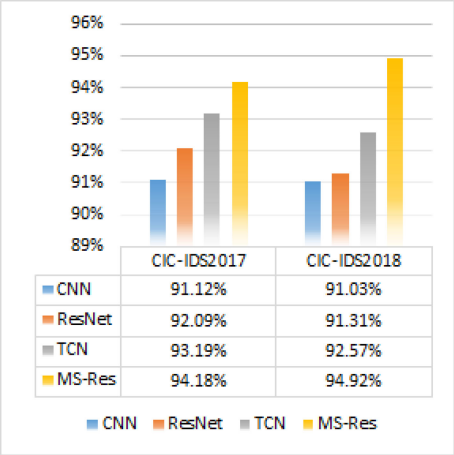

Fig. 5. Impact of TA module on the model's F1-measure.

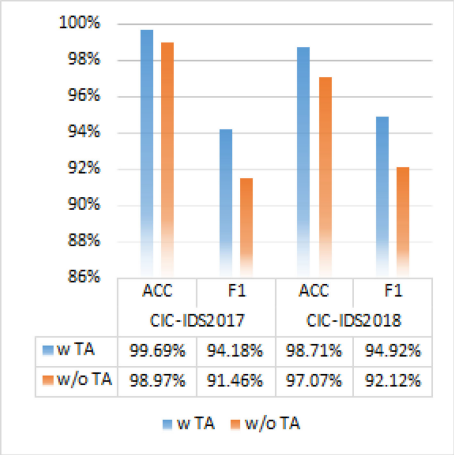

Fig. 6. Impact of TA module on the model's performance.

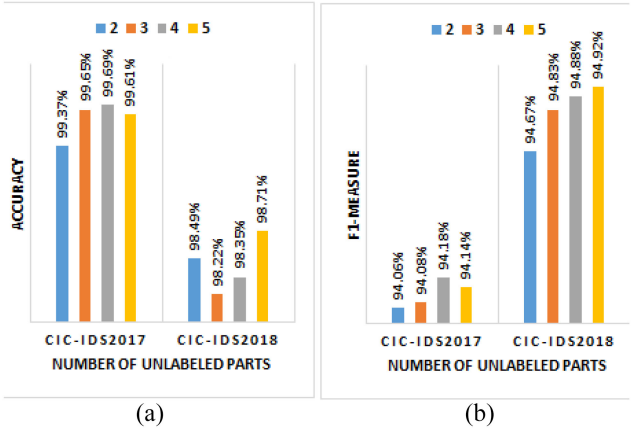

Fig. 7. Impact of number of parts of unlabeled data set on the (a) model's accuracy and (b) model's F1-measure.

TABLE IX
COMPUTATIONAL TIME FOR BOTH DATA SETS

	CIC-IDS2017			CIC-IDS2018		
	Cloud		Fog	Cloud		Fog
Study	Train time (s)	Test time (s)	Test time (s)	Train time (s)	Test time (s)	Test time (s)
RF	171	36.9	57.2	351	35.5	60.3
LSTM	82	23.7	68.9	234	24.0	71.4
ANN	46	1.4	3.8s	302	1.9	5.1
AE + ANN	108	2.8	7.1	394	2.9	8.9
DBN + ANN	354	2.2	6.7	1492	2.5	6.7
Proposed	79	1.1	3.2	213	1.2	3.4

inclusion of TA module improves the performance with 0.7%–2% and 2%–3% improvements on accuracy and F1-measure, respectively. That explains the role of TA in forcing the model to attend to more important features during the learning process.

3) Impact of Number of Unlabeled Parts of Data: This experiment was carried out to decide the optimal number of parts P in the unlabeled set of data. Hence, we evaluate the performance of the proposed SS-Deep-ID using different values of P starting from 2 to 5 to maintain reasonable computational complexity. where the corresponding performance in terms of accuracy and F1-measure is presented in Fig. 7. It could be noted that the CIC-IDS2017 achieve optimal performance when $P = 4$, meanwhile, the CIC-IDS2018 achieve optimal accuracy and F1-measure when $P = 5$. This validates the choice of the number of labeled parts and indicates its sensitivity for the characteristics of data.

E. Computational Analysis

Owing to the resource-constrained nature of IoT environments and the time-sensitivity of IDS, this section analyzes the computational cost required to attain the optimal performance, we compare the model's training time and inference time (for a batch of 1 000 000 records). Since the SS-Deep-ID is deployed on the fog layer of IoT network, we experiment the inference time on a fog device established using a Dell laptop [i.e., Intel Core i5-3317U CPU @ (1.7 GHz) processor with 8-GB RAM Windows 10 64-bit and 500-GB HDD]. The time values of the competing approaches on both data sets are shown in Table IX. For supervised approaches, it can be seen that ANN has the shortest training time on both data sets. On the other hand, LSTM, SVM, and RF have the longest training times. The semi-supervised approaches (i.e., TSVM [36], DBF+ANN [15], and AE+ANN) take longer to complete the training task owing to the costly computation consumed through the unsupervised pretraining phase. The proposed SS-Deep-ID achieved comparatively the shortest training time. Similar behavior can be observed with inference time, where ANN attained the fastest inference and LSTM and RF achieved the slowest inference, with the proposed model having the fastest inference among all semi-supervised approaches. Besides, repeating these experiments again on the fog device show a great increase for all models, however, the proposed SS-Deep-ID maintains achieving the lowest inference time on both data sets.

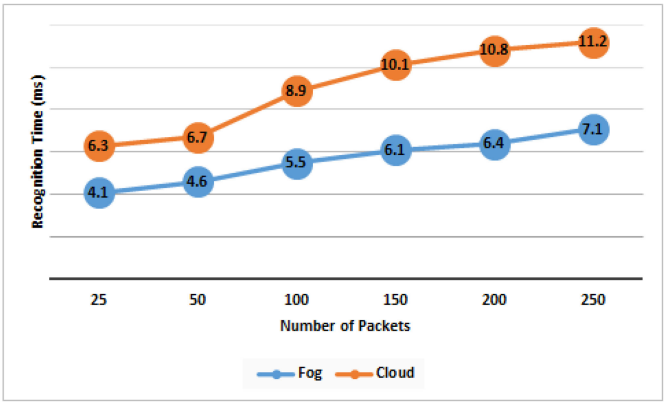

Fig. 8. Memory and CPU overhead for SS-Deep-D on the fog node.

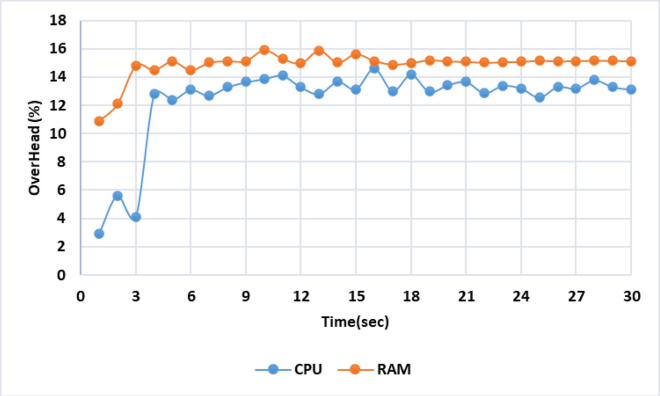

Fig. 10. Memory and CPU overhead for SS-Deep-D on the fog node.

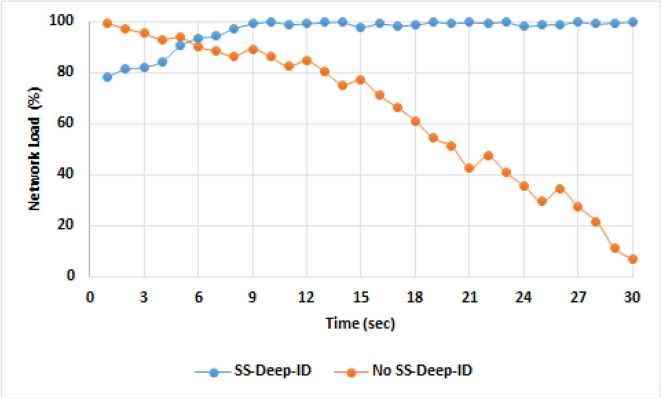

Fig. 9. Network load in the presence and absence of SS-Deep-D on the fog node.

This is explained by several factors, first, the stacked CD-Conv enables having a wider receptive field using just a few layers, meanwhile maintaining the computational efficiency. Which in turn lessen the number of trainable parameters, producing light-weight architecture with efficient training. Besides, the BN layers and residual connection speed the computation and ease the gradient flow through training. Moreover, splitting the unlabeled data into different parts enable help speeding up the training process at later parts. Furthermore, after experimentation with different numbers of training epochs, the proposed model has an optimal performance when the training epochs are 10 compared with 50 epochs for other semi-supervised approaches which in turn indicates the rapid convergence of the model. According to the above discussions, the proposed SS-Deep-ID is computationally efficient to deployed in the IoT environment. It could be easily trained on a centralized cloud server where a large amount of labeled and unlabeled IoT traffic records is aggregated and prepared for training. The trained version of SS-Deep-ID can be reliably installed on fog or edge nodes to promote real-time intrusion detections in different parities of IoT networks.

F. Network Analysis

1) Recognition Time: Since the recognition or response time is a critical factor for any real-time IoT system, it is inevitable to assess the performance of the proposed SS-Deep-ID under two IoT deployment configurations, namely, cloud-based and fog-based deployment. Fig. 8 displays the time was undertaken by the proposed SS-Deep-ID to act in response to a different number of packets of IoT traffic (from CIC-IDS2018 data) for both configurations. It is notable that the fog deployment realizes a short response time compared to the cloud deployment using both data sets. The justification for this short time is because of both edge detection and the increased processing power of our fog node. Thus, whatever the selected fog node or batch size, the fraction of the time remains smaller than the taken by cloud. The primary reason for the longer response time incurred by the cloud is owing to the communication delay and a large amount of data needs to be received and processed from all fog nodes. Therefore, as expected the fog deployment is recommended for the proposed SS-Deep-ID.

2) Network Load: In addition to the recognition time, the network load is an important aspect for determining the efficiency of any fog-based IoT framework. Accordingly, the additional experiment is performed to measure the network using load multi router traffic grapher software.[4] Fig. 9 shows the network load throughout the intrusion when the SS-Deep-ID is installed and not installed (i.e., cloud-based) on the fog layer. It could be seen that deploying the SS-Deep-ID helps to reduce the load of the network at the time the intrusion is executed. This further explains the robustness of SS-Deep-ID at the fog layer.

3) Memory and CPU Overhead: The cloud environment is commonly known to have rich and powerful memory and CPU resources. However, fog nodes did not possess this capability. Thus, experimenting with the memory and CPU overhead of SS-Deep-ID is an essential aspect to be investigated. For this objective, Fig. 10 displays the measurements concerning the CPU and memory usage throughout the execution of the SS-Deep-ID on our fog device for 30 s. For memory overhead, the fog device initially has 11% of allocated memory before

[4]https://oss.oetiker.ch/mrtg/

the experiment starts. Once the experiment starts, the memory overhead raised to 14%–15%. then remain fluctuating between 15% and 16% till the end of time, which indicates 5% of the average increase in memory usage. For CPU overhead, the fog device initially has 3% of the busy CPU before the experiment starts. By the beginning of the third experiment, the CPU overhead increased and remain fluctuating between 12% and 15% till the end of time, which indicates 10% of the average increase in the CPU usage.

VI. Limitations

Despite the efficiency and effectiveness of the proposed model, it is not clear whether it can preserve its effectiveness in the scenario of a huge amount of IoT traffic data. Additionally, a variety of techniques might have liabilities for various data properties, which is not been investigated for the proposed model. Furthermore, distributed training is an important challenge for most intelligent IoT applications, which is not addressed in this study but will be addressed in future work as an expansion of the model. Moreover, the privacy-preservation aspect of the proposed SS-Deep-ID has not been considering and federated learning could be exploited to address this aspect in the future. Finally, interpretability and uncertainty of results are not addressed in this work, which is a promising challenge in many IoT applications.

VII. Conclusion

This article presents a novel DL architecture, called SS-Deep-ID, for intrusions/cyber-attacks detection from IoT traffic records. We employ DC-Conv to mitigate the drawbacks of standard C-Conv and a novel multiscale residual temporal convolution block is proposed to improve the network capability in capturing spatiotemporal representations (i.e., interrelationships) from input traffic sequences. Additionally, an attention module is introduced to assist the proposed SS-Deep-ID to focus on important information during training. Furthermore, the SS-Deep-ID model is trained in a semi-supervised manner with labeled and unlabeled IoT traffic records. The comprehensive evaluation has shown the superiority of the proposed SS-Deep-ID across different measures and has also validated its computational effectiveness. This qualifies the SS-Deep-ID to be used as a general IDS in a wide variety of IoT applications.

VIII. Future Work

This study opens the way for several future improvements.

1) *Immediate Response:* Offering real-time decisions has become an essential requirement for the mainstream of IoT applications (i.e., smart transportation or smart industry), so real-time detection of intrusions/attacks is of great importance. The intrusion detector could cause a great failure in the underlying system when it consumes a longer period in manipulating a sequence of IoT traffic than the time between received traffics.
2) *Online Adaptive Learning:* The inconsistent nature of IoT time-series traffic necessitates the development of an adaptive DL technique for detecting intrusions. Thus, even though offline techniques are potentially used for early model deployment, there should be some techniques that enable the DL model to evolve through time to become familiar with expected and unexpected variations in the distributions of IoT traffic without necessitating complete retraining for the model.
3) *Generalization:* Considering the fact that there is no specific method to perform well in all circumstances, improving the performance of DL-based intrusion detection models across different data sets will offer valuable information for improved decisions making un multidomain scenario, increase model reusability, and facilitate the corresponding deployment for the diversity of applications.
4) *Explainability and Interpretability:* Research on the interpretability of decisions taken by ML and DL approaches needs to be explained and validated. However, the interpretability of DL-based IDS remains an unexplored area that needs to be investigated in future studies.

References

[1] Z. Lv, L. Qiao, J. Li, and H. Song, "Deep learning enabled security issues in the Internet of Things," *IEEE Internet Things J.*, early access, Jul. 9, 2020, doi: 10.1109/JIOT.2020.3007130.

[2] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9042–9053, Oct. 2019.

[3] A. Cook, G. Mısırlı, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.

[4] C. Pu, "Sybil attack in RPL-based Internet of Things: Analysis and defenses," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4937–4949, Jun. 2020, doi: 10.1109/JIOT.2020.2971463.

[5] L. Fang, Y. Li, Z. Liu, C. Yin, M. Li, and Z. J. Cao, "A practical model based on anomaly detection for protecting medical IoT control services against external attacks," *IEEE Trans. Ind. Informat.*, early access, Jul. 23, 2020, doi: 10.1109/TII.2020.3011444.

[6] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart., 2020, doi: 10.1109/COMST.2020.2988293.

[7] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the Internet of Things (IoT) forensics: Challenges, approaches, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1191–1221, 2nd Quart., 2020, doi: 10.1109/COMST.2019.2962586.

[8] M. M. Hassan, S. Huda, S. Sharmeen, J. Abawajy, and G. Fortino, "An adaptive trust boundary protection for IIoT networks using deep-learning feature-extraction-based semisupervised model," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2860–2870, Apr. 2021, doi: 10.1109/TII.2020.3015026.

[9] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K.-K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8852–8859, Sep. 2020, doi: 10.1109/JIOT.2020.2996425.

[10] L. Li, J. Yan, H. Wang, and Y. Jin, "Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 13, 2020, doi: 10.1109/TNNLS.2020.2980749.

[11] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Fault-attention generative probabilistic adversarial autoencoder for machine anomaly detection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7479–7488, Dec. 2020, doi: 10.1109/TII.2020.2976752.

[12] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020, doi: 10.1109/COMST.2020.2970550.

[13] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3242–3254, Mar. 2021, doi: 10.1109/JIOT.2020.3002255.

[14] Y. Liu *et al.*, "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, early access, Jul. 24, 2020, doi: 10.1109/JIOT.2020.3011726.

[15] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102767.

[16] Y. An, F. R. Yu, J. Li, J. Chen, and V. C. M. Leung, "Edge intelligence (EI)-enabled HTTP anomaly detection framework for the Internet of Things (IoT)," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3554–3566, Mar. 2021, doi: 10.1109/JIOT.2020.3024645.

[17] W. Li, W. Meng, and M. H. Au, "Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments," *J. Netw. Comput. Appl.*, vol. 161, Jul. 2020, Art. no. 102631.

[18] P. F. de Araujo-Filho, G. Kaddoum, D. R. Campelo, A. G. Santos, D. Macêdo, and C. Zanchettin, "Intrusion detection for cyber-physical systems using generative adversarial networks in fog environment," *IEEE Internet Things J.*, early access, Sep. 18, 2020, doi: 10.1109/JIOT.2020.3024800.

[19] M. Abdel-Basset, V. Chang, H. Hawash, R. K. Chakrabortty, and M. Ryan, "Deep-IFS: Intrusion detection approach for IIoT traffic in fog environment," *IEEE Trans. Ind. Informat.*, early access, Sep. 22, 2020, doi: 10.1109/TII.2020.3025755.

[20] L. Vu, V. L. Cao, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Learning latent representation for IoT anomaly detection," *IEEE Trans. Cybern.*, early access, Sep. 18, 2020, doi: 10.1109/TCYB.2020.3013416.

[21] N. Ravi and S. M. Shalinie, "Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3559–3570, Apr. 2020, doi: 10.1109/JIOT.2020.2973176.

[22] Y. Cheng, Y. Xu, H. Zhong, and Y. Liu, "Leveraging semisupervised hierarchical stacking temporal convolutional network for anomaly detection in IoT communication," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 144–155, Jan. 2021, doi: 10.1109/JIOT.2020.3000771.

[23] D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu, and R. Li, "LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5244–5253, Aug. 2020, doi: 10.1109/TII.2019.2952917.

[24] J. Gao *et al.*, "Omni SCADA intrusion detection using deep learning algorithms," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 951–961, Jan. 2021, doi: 10.1109/JIOT.2020.3009180.

[25] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6882–6897, Aug. 2020, doi: 10.1109/JIOT.2020.2970501.

[26] M. Mahbub, "Progressive researches on IoT security: An exhaustive analysis from the perspective of protocols, vulnerabilities, and preemptive architectonics," *J. Netw. Comput. Appl.*, vol. 168, Oct. 2020, Art. no. 102761.

[27] B. Wang, Y. Sun, and X. Xu, "A scalable and energy-efficient anomaly detection scheme in wireless SDN-based mMTC networks for IoT," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1388–1405, Feb. 2021, doi: 10.1109/JIOT.2020.3011521.

[28] H. Yang, S. Liang, J. Ni, H. Li, and X. S. Shen, "Secure and efficient k NN classification for industrial Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10945–10954, Nov. 2020, doi: 10.1109/JIOT.2020.2992349.

[29] N. Ravi and S. M. Shalinie, "Semisupervised-learning-based security to detect and mitigate intrusions in IoT network," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11041–11052, Nov. 2020, doi: 10.1109/JIOT.2020.2993410.

[30] T. Ergen and S. S. Kozat, "Unsupervised anomaly detection with LSTM neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3127–3141, Aug. 2020, doi: 10.1109/TNNLS.2019.2935975.

[31] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Trans. Ind. Informat.*, early access, Sep. 11, 2020, doi: 10.1109/TII.2020.3022432.

[32] *CSE-CIC-IDS2017 Dataset*, CIC, Fredericton, NB, Canada, 2017. Accessed: Oct. 4, 2020. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[33] *CSE-CIC-IDS2018 Dataset*, CIC, Fredericton, NB, Canada, 2018. Accessed: Oct. 4, 2020. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html

[34] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. H. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.

[35] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 464–471, Mar. 2002.

[36] X. Wang, J. Wen, S. Alam, Z. Jiang, and Y. Wu, "Semi-supervised learning combining transductive support vector machine with active learning," *Neurocomputing*, vol. 15, no. 3, pp. 1288–1298, 2016.

[37] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2017, pp. 5998–6008.

[38] P. Virtanen *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nat. Methods*, vol. 17, pp. 261–272, Feb. 2020.

[39] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?" *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 41–49, Sep. 2018.

[40] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to Pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1472–1514, 3rd Quart., 2020, doi: 10.1109/COMST.2020.2965856.

[41] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1686–1721, 3rd Quart., 2020, doi: 10.1109/COMST.2020.2986444.

[42] A. Achille and S. Soatto, "Information dropout: Learning optimal representations through noisy computation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2897–2905, Dec. 2018.

[43] S. Wu *et al.*, "$L1$-norm batch normalization for efficient training of deep neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 2043–2051, Jul. 2019.

[44] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. B. Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020.

[45] Z. Karimi, M. M. R. Kashani, and A. Harounabadi, "Feature ranking in intrusion detection dataset using combination of filtering methods," *Int. J. Comput. Appl.*, vol. 78, no. 4, pp. 21–27, Sep. 2013.

[46] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems," *Int. J. Eng. Technol.*, vol. 7, no. 24, pp. 479–482, 2018.

[47] M. Di Mauro, G. Galatro, and A. Liotta, "Experimental review of neural-based approaches for network intrusion management," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2480–2495, Dec. 2020.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.