Deep-IFS: Intrusion Detection Approach for IIoT Traffic in Fog Environment

Mohamed Abdel-Basset, Victor Chang, Hossam Hawash, Ripon K. Chakrabortty, and Michael Ryan

***Abstract*— The extensive propagation of Industrial Internet of things (IIoT) technologies has encouraged the intruders to initiate a variety of attacks that need to be identified to maintain the security of end-user data and the safety of services offered by service providers. Deep learning (DL), especially recurrent approaches, has been applied successfully to the analysis of IIoT forensics but their key challenge of recurrent DL models is that they struggle with long traffic sequences and can't be parallelized. Multi-head Attention (MHA) tried to address this shortfall but fails to capture the local representation of IIoT traffic sequences. In this paper, we propose a forensics-based DL model (called Deep-IFS) to identify intrusions in IIoT traffic. The model learns local representations using local gated recurrent unit (LocalGRU), and introduces an MHA layer to capture and learn global representation (i.e., long-range dependencies). A residual connection between layers is designed to prevent information loss. Another challenge facing the current IIoT forensics frameworks is their limited scalability, limiting performance in handling Big IIoT traffic data produced by IIoT devices. This challenge is addressed by deploying and training the proposed Deep-IFS in a fog computing environment. The intrusion identification becomes scalable by distributing the computation and the IIoT traffic data across worker fog nodes for training the model. The master fog node is responsible for sharing training parameters and aggregating worker nodes output. The aggregated classification output is subsequently passed to the cloud platform for mitigating attacks. Empirical results on the Bot-IIoT dataset demonstrate that the developed distributed Deep-IFS can effectively handle Big IIoT traffic data compared with the present centralized DL-based forensics techniques. Further, the results validate the robustness of the proposed Deep-IFS across various evaluation measures.**

***Index Terms*—Deep Learning, Industrial Internet of Things, Forensics, Intrusion Detection**

I. Introduction

THE Industrial Internet of Things (IIoT) is a ubiquitous network with a wide span of interconnected smart devices that afford a variety of intelligent computing amenities in the industrial environment. For instance, IIoT nodes can discern, handle, and transfer a diversity of information into and out of IIoT platforms to offer efficient services and user experience in domains varying from manufacturing to service provision. In the opposite direction, IIoT infrastructures and devices can also be invaded by attackers seeking to attain critical information and promote outrageous actions [1].

In order to alleviate the hazard of intrusions on IIoT devices, effective Intrusion Detection System (IDS) are required to constantly screen IIoT flows that originate from a variety of sources, and scrutinize them to recognize indications of possible intrusions /cyber-attacks [2], [3]. IDS can be categorized into two groups: signature-dependent and anomaly-dependent. Signature-dependent IDS identify malicious interventions (or doubtful actions) by analyzing flows based on formerly learned rules of acknowledged attacks. However, signature-dependent IDS have a number of problems. First, they can only recognize previously identified intrusions with well-examined features. They fail to identify new and unseen threats, which is problematic since intruders continuously improve their techniques, changing intrusion actions to evade outdated security measurements [2]. Second, the rise in the count newly acknowledged intrusions/ cyber-attacks significantly increases the count of signatures, causing extra comparisons between stored outlines and received actions. This increases the IDS workload, explicitly impacting the responsivity of the IDS resulting in a critical problem for immediate detection of intrusions. Consequently, such IDS usually exclude scrutiny of proceedings at particular rates based on accessible processing capabilities [3]. Third, IDS often necessitate intervention from human experts to investigate, analyze, and interpret signatures for the novel intrusions, and it could take a year to examine the attributes of a certain intrusion or cyber-attack [4].

The before-mentioned problems have been investigated using anomaly-dependent IDS that monitor a sequence of received IIoT traffic and the trained model distinguishes anomalies according to the resemblance index between standard and anomalous remarks. The key challenge of such systems is to learn an exceptional normal event that comprises numerous underlying changing activities produced by separate sources of IIoT traffic. Various kinds of IIoT sources could produce various events raising the false-positive ratio due to lower resemblance between received events and the normal learning activities [5].

Recently, there has been increasing research interest in investigating the usefulness of artificial intelligence (AI) approaches in developing cybersecurity methods, such as privacy-preservation [6], threat prediction [7], malware disclosure [8], and

M. Abdel-Basset and **H. Hawash** are with the Department of Computer Science, Zagazig University, Zagazig, 44519 Egypt. (mohamed.abdelbasset@fci.zu.edu.eg; hossamreda@zu.edu.eg)

V. Chang with Teesside University, Middlesbrough, UK TS1 3BA (victorchang.research@gmail.com)

R. K. Chakrabortty and M. Ryan are with the Capability Systems Centre, School of Engineering and IT, UNSW Canberra; Australia, (r.chakrabortty@adfa.edu.au; m.ryan@adfa.edu.au)

forensic exploration [1]. Among AI approaches, deep learning (DL) introduces improved learning models that have been achieving great success in various domains of computations, including IIoT based systems. According to the adopted learning scheme, these models are usually divided into three groups: supervised models, unsupervised models, and semi-supervised models [9]. The supervised models (convolutional neural network (CNN), recurrent neural network (RNN)) are trained to maps the features of IIoT flow to the corresponding class, i.e., benign or malicious, and they are trained using labeled data. On the other hand, the unsupervised models (autoencoder (AE) and deep belief network (DBN)) are trained to learn from the unlabeled features of IIoT flow by learning the informative patterns the IIoT data. Still, they often need substantial amounts of data and often computationally exhaustive [13]. Hence, semi-supervised models emerged as a combination of supervised and unsupervised models and are trained to learn from both labeled and unlabeled data to classify unlabeled data.

The present approaches for detecting intrusions rely chiefly on a centralized cloud, which means that they cannot cope with current IIoT requirements, such as the distribution of computation, fast response, and scalability [14]. In IIoT environments, requests and computations are initiated at a large number of IoT devices, which produce vast amounts of traffic data. Centralized cloud computing is vitally essential in IIoT since it permits individuals to access and use various IIoT services accessible through the internet. However, due to its reliance on centralized computation, it is unable to process the traffic effectively from IIoT devices while carrying out exhaustive computations [15]. Moreover, long recognition times are experienced due to the extended distance between the IIoT devices and the centralized intrusions/cyber-attack recognition system, causing a long communication time [16]. Consequently, an evolving distributed IIoT computation called fog computing (FC) is employed to overcome these limitations. The primary notion behind FC is to position data processing/ storage closer to the IIoT traffic sources. Thus, the fog layer is used to deploy the intelligent security technique since it contains several fog nodes that enable the distribution of the computation [17]. Therefore, expensive storage and processing incurred by the IIoT devices could be reduced by the deployment of distributed IDS [18]. Accordingly, we propose to develop a novel distributed DL framework to detect IIoT intrusions FC environment.

To summarize, the main challenges addressed in this study are summarized as follows. First, current ML techniques have shown unstable and significantly affected by the nature of data and the feature engineering employed. Second, the present DL approaches either use CNN or recurrent networks. However, CNN fails to capture the long-term characteristics of IIoT traffic data. Additionally, recurrent networks suffer from the problem of gradient explosion and vanishing, which chiefly restrict their capability to capture long-term patterns; their sequential characteristics make them tremendously difficult, if not impossible, to be parallelized during the execution. Although the MHSA mechanism tried to address some of the limitations of RNNs, it still suffers from positional information loss since it deals with every position equally, and it disregards the local representation of IIoT traffic that is intrinsically significant. Third, centralized cloud computing-based IDS suffer from limited scalability, latency, and expensive computations, meaning that they exhibit a long detection time and are ineffective in handling enormous IIoT traffic data.

In view of this, we propose the Deep-IFS model that combines the benefits of RNNs and MHSA and simultaneously overcomes their limitations. The LocalGRU was proposed to extract and learn local representations from the normalized IIoT traffic. The MHSA layer is followingly introduced to capture long-term information. The residual "add and normalize" connection is employed to avoid losing information from one layer to another. After that, a feed-forward layer is proposed to perform a non-linear feature transformation. To overcome the shortcomings of centralized training, we propose to train the Deep-IFS in a distributed manner in a fog computing environment.

The main contributions of this work are as follows.

- We propose a novel DL model, called Deep-IFS, to overcome the limitations of meeting the current DL approaches for detection of intrusions and cyber-attacks.
- We employ Local GRU to enable efficient extraction of local representation from IIoT traffic.
- We capture the long-term dependencies using the MHSA layer, which enables distributing/parallelizing the execution of the model.
- We distribute the learning of Deep-IFS across different fog computing nodes to mitigate the drawbacks of centralized learning.

The remainder of the paper is as follows. Section II introduces the research literature relevant to IIoT-based cyber-attacks, IDS for IIoT environment, DL-dependent IDSs, DL for big data analytics, and DL for the fog environment. Section III describes the proposed Deep-IFS for intrusion detection in IIoT networks. Section IV introduces the experimental configurations, results, comparisons, analysis, and limitations. Finally, conclusions and further work are presented in section V.

II. Related Studies

In this section, we introduce the studies related to this work. Initially, we present the recent work related to IIoT-based cyber-attacks. Then we discuss the recent approaches for IIoT intrusion/anomaly detection, including some of the most recent and relevant DL approaches. We then present the challenges presented in recent literature to IIoT environments. Finally, we present recent fog computing studies that addressed these challenges.

A. Digital forensic in the Internet of Things

Digital forensics (DF) is a group of methods developed by researchers to investigate, analyze, and identify the adversaries to protect the critical information processed and transferred through IIoT networks. Over time, and owing to continuous technological and industrial improvements, the DF has been advanced into numerous subclasses, each concentrating on events of occurring in diverse settings, specifically: IoT forensics, mobile forensics, memory forensics, data forensics, and cloud forensics [29].

Table I. Summarization of the recent cutting-edge study for IIoT intrusions/Cyber-attacks detection.

Study (year)	Year	Security Paradigm	Algorithm Novelty	Model	Feature selection	Dataset	Significance test
Shafiq et al. [28]	2020	Centralized	Standard	a bijective soft set+ 5 ML algorithms	Yes	One Dataset	None
Koroniotis et al. [19]	2020	Centralized	Novel	DL + particle swarm optimization	No	Two datasets	None
Alkadi et al. [30]	2020	Distributed	Standard	Bi-LSTM	No	Two datasets	None
Saharkhizan et al. [4]	2020	Centralized	Novel	Ensemble of LSTMs + decision tree	No	One Dataset	None
Wu et al. [7]	2019	Centralized	Novel	LSTM + Gaussian Naïve Bayes	No	Three Datasets	None
Ferrag et al. [13]	2020	Centralized	Standard	DNN, CNN, RNN	No	Two datasets	None
Koroniotis et al. [20]	2019	Centralized	Standard	SVM, RNN, and LSTM	Yes	Two datasets	None
Amma et al. [18]	2020	Distributed	Standard	Vector convolutional DL (VCDL)	No	One dataset	None

This study focuses on IoT forensics (IFS), which addresses security events in IoT networks, commonly using logs and acquired packages to identify intrusions/cyber-attacks. As argued in the previous section, several IDS have been introduced [11,12,30]. However, none has emerged as the favored technique by specialists, since calibration is missed, and differing state of affairs necessitates various tools and techniques [2,31,32].

B. Intrusion Detection Approaches for IIoT environment

The current learning approaches for IIoT forensics are comprehensively surveyed in [2], which investigated the key challenges facing IIoT security and IDS, including the learning techniques by intruders, the privacy of the learning techniques, and the structure of these techniques. Machine learning (ML) approaches (i.e., extreme learning machine (ELM), regression, k-means, support vector machine (SVM), and the Bayesian network) have been employed for identifying and detecting IIoT anomalies and intrusions [10]. These intelligent approaches are able to deliver appropriate solutions for detecting intrusions/cyber-attacks with the respectable performance [14]. However, ML approaches have several limitations. First, their performance relies heavily on the robustness of the employed feature engineering technique, limiting their stability. Second, their performance worsens when applied to big and high-dimensional data. Third, the learning capabilities are not robust enough to cope with the dynamic nature of data (cyber-attacks) in the IIoT environment [16].

Therefore, DL has emerged as a novel learning paradigm to address the before-mentioned limitations due to its powerful learning capabilities (especially from high-dimensional data), adaptivity to dynamic environments, and independence from any feature engineering. DL techniques have been demonstrated to have the intrinsic ability to alleviate and resolve the issues associated with conventional approaches [23]. A variety of DL approaches have been applied in IDS such as CNN [13], RNN [20], DBN [19,] and longest-short-term-memory (LSTM) [20]. For example, Ferrag et al. [13] investigated CNN, RNN, and DNN for intrusion detection and completed a comparative analysis of their performance under different configurations. Amma et al. [18] introduced vector convolution to build IDS in fog based IoT. However, CNN usually fails to sequential IoT streams, especially those that they have a long-term dependency. Hence, Sahakian et al. [4] proposed to use the LSTMs to alleviate this problem, where an ensemble of LSTMs employed to act as detectors and their output were merged into a decision tree for final classification. Similarly, Alkadi et al. [30] developed IDS using bidirectional LSTM (Bi-LSTM) that are integrated into the blockchain-enabled system. However, the computational cost of these models is high. To address this, Liaqat et al. [5] proposed a novel framework that integrates CNN and Cuda LSTM (cuDNNLSTM) to timely and effectually detect complex malware botnets in the medical IoT environment.

To summarize the recent research findings in IDS, Table I present the most relevant studies for IDS, including their reference, publication year, security paradigm (i.e., centralized or distributed processing), the novelty of algorithm (i.e., standard DL, hybrid model, novel model). Whether they use feature selection or not, the DL techniques employed, the number of datasets used, and the statistical test used in each study.

C. DL For Big Data Analytics

The recent and continuing development of smart computing (i.e., smart devices) has produced massive volumes of data that lead to big data analytics requirements. However, the multiplicity of big data opportunities in commercial applications, smart industries, smart healthcare, and intelligent transportation, big data processing remains a challenging task because of its characteristics, including huge volume and multi-dimensionality velocity, diversity, and reliability [23]. DL approaches have shown to effectively extract features, recognize important patterns, learn representative information from the massive data volumes [24].

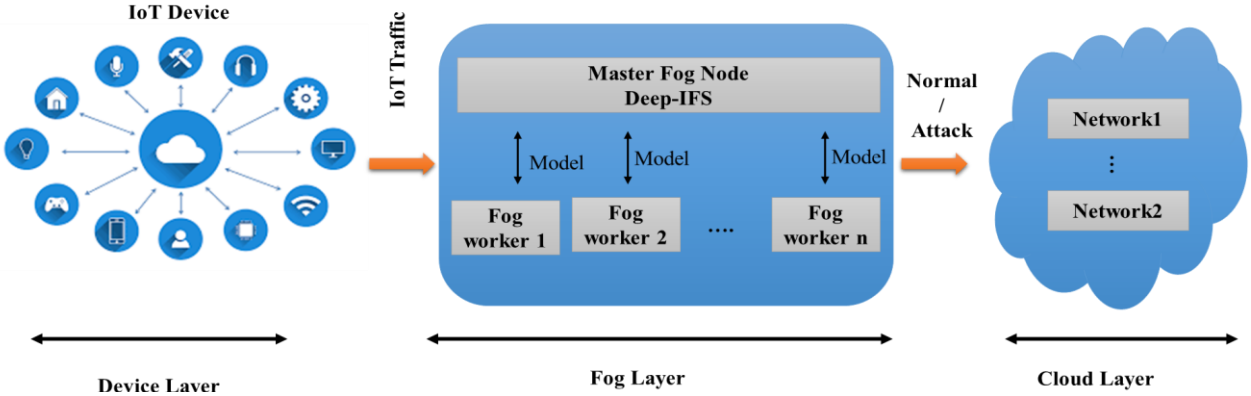

Fig. 1. Systematic diagram of the proposed IIoT forensics framework

Thus, DL could be broadly employed to effectively resolve big data challenges, which is not possible using traditional ML techniques since Significant human participation is mandatory for developing efficient techniques [23]. Fine-grained data representation has been exposed by the DL models to produce timely and precise recognitions; nevertheless, the most critical challenges occur in computation distribution and computational scalability [24]. Thus, fog computing (FC) has emerged to offer distributed learning from big data.

D. DL For Fog environment

Recently, several studies reviewed and investigated different learning techniques in the fog environment. The main target of FC is to alleviate the workloads of the IIoT network at the cloud by bringing the storage and the complex computations closer to the client devices. To tackle the scalability problem incurred during learning from big traffic data originated from different IIoT-enabled devices, fog nodes are employed to act as a substituting computational unit. DL approaches can be exploited to demonstrate the viability of deploying intelligent learning paradigms in the fog environment. Furthermore, the learning and the inferencing time could be minimized through parallel and distributed processing of different IIoT forensics. Therefore, we propose a novel DL approach (Deep-IFS) to analyze and recognize anomalies/intrusions in IIoT traffic presented in an FC environment.

III. Proposed Approach

The IIoT network encompasses a large number of IIoT devices that are could be located in diverse places. Thus, intrusion detection techniques might be skilled enough to learn from the generated traffic data by these devices to offer a reliable response in for user's requests within a short time. In view of this, the centralized approach of IIoT forensics has shown inefficient detection time and accuracy. The proposed Deep-IFS is developed to process the traffic flow data from smart IIoT devices by dividing the workload across different worker FC nodes. Fig.1 shows the developed IIoT forensics approach, including the device layer, fog layer, and the cloud layer. The device layer contains the smart IIoT devices interconnected with each other. It has limited computational power and restricted bandwidth; thereby, it fails to handle emerging actions. The fog layer encompasses the tools and devices required to constitute the connections of the IIoT network. It is accountable for minimalizing computational overhead on the IIoT devices that have limited resources. The cloud layer is responsible for authenticating the information attained from the fog. It offers guiding principles to the fog layer for improving the eminence of services (i.e., responses) afforded the fog node. The proposed Deep-IFS for IIoT forensics accomplish distributed training by scattering the computation evenly on the fog nodes. Specifically, the proposed Deep-IFS is shared between the worker fog nodes. The master node, the worker nodes employed to learn to discriminate between the IIoT traffic as either benign or intrusion and followingly passed for supplementary computation on the cloud layer.

A. The representation of IIoT traffic

The traffic sample of the IIoT network is represented as $TS = [tf_1, tf_2, \cdots, tf_n, C_l]$, where td_i denote the $i-th$ the traffic features and the C_l represented the corresponding the label of the IIoT traffic sample. Hence, the whole dataset of IIoT traffic can be denoted with equation (1).

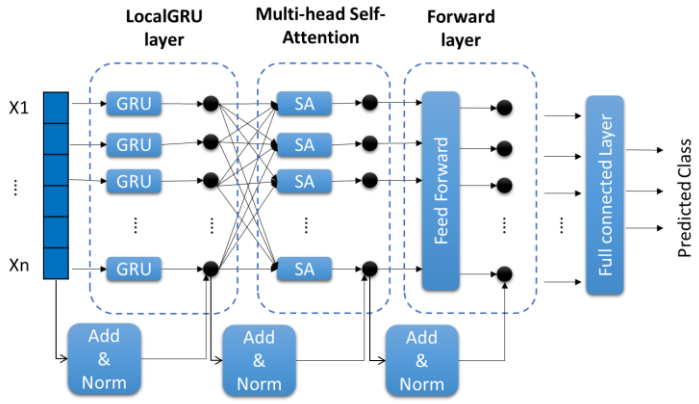

Fig. 2. The architecture of proposed Deep-IFS for intrusion detection in IoT.

$$TD = \begin{vmatrix} tf_1^1 & tf_2^1 & \cdots & tf_n^1 \\ tf_1^2 & tf_2^2 & \cdots & tf_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ tf_1^k & tf_2^k & \cdots & tf_n^k \end{vmatrix} \quad (1)$$

where n *and* k denote the number of features and the number of samples, respectively. The $min-max$ normalization is applied to normalize the feature into the interval $[0, 1]$ according to the equation (2).

$$X_i^k = \frac{tf_i^j - min(TS^j)}{max(TS^j) - min(TS^j)} \quad (2)$$

where tf_i^j represents the $i-th$ feature in $j-th$ sample. The $max(TS^j)$ and the $min(TS^j)$ represent the minimum and maximum value of the $j-th$ sample, respectively. So, the normalized IIoT traffic dataset can be represented by equation (3).

$$TD_{normalized} = \begin{vmatrix} X_1^1 & X_2^1 & \cdots & X_n^1 \\ X_1^2 & X_2^2 & \cdots & X_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ X_1^k & X_2^k & \cdots & X_n^k \end{vmatrix} \quad (3)$$

The normalized data (3) is fed into the proposed Deep-IFS to learn to detect intrusions/ cyber-attacks from the network traffic.

B. The proposed Deep-IFS Approach

Since RNNs fail to acquire long-standing dependencies and are unable to accomplish parallel computation on sequential/temporal data (i.e., traffic data), recent studies proposed to eschew the recurrent model in favor of the attention technique [25]. Because a multi-head self-attention mechanism (MHSA) enables capturing interrelated positional information, it offers a flexible flow of information without incurring any loss. Nevertheless, the MHA still suffers from ignoring the imperative local patterns (representation) [26]. Motivated by the transformer network [25], we propose Deep-IFS to combine RNN and MHA benefits while evading the before-mentioned limitations. The Deep-IFS comprises of three layers, as revealed by Fig.2. The lowermost part is proposed to learn local traffic information using gated recurrent unit (GRU), which is well known for its robustness for sequential learning and also more time-efficient compared to the other recurrent DL models (i.e., LSTM) [6-7]. The intermediate layer is formed using the MHSA layer. The topmost layer is responsible for performing non-linear feature transformation.

The local GRU systematizes the incoming long traffic sequence into a shorter sub-sequence that contains the local representation that is individualistically and conformably handled by a common GRU. The Deep-IFS constitutes a local window with width *M* for

each targeted traffic, which comprises M successive positions and ends at the target site. Consequently, local GRU just emphasizes local short-range dependencies. Figure 2 shows a schematic of the local RNN. Uniquely, the position of a local sub-sequences of length M is specified as $M = [x_{t-M-1}, x_{t-M-2}, \cdots, x_t]$, which are handled by the local GRU to produce a vectorized representation of M hidden states, where the latter hidden vector is epitomized as a feature of the local sub-sequence.

$$h_t = LocalGRU(x_{t-M-1}, x_{t-M-2}, \cdots, x_t) \quad (4)$$

Thus, the local GRU slides each window one at a time and subsequently links the representation of each local area as a vectorized representation of the local hidden state of the whole traffic sequences.

$$h_1, h_2, \cdots h_n = LocalGRU(x_1, x_2, \cdots x_n) \quad (5)$$

After that, the vector of hidden state representation is fed into the MHSA layer to extract and learn the long-term reliance patterns from the input traffic sequence. While having the present vector representation, the computation of the output of the MHSA layer is computed according to the equation (6).

$$u_t = MHSA(h_1, h_2, \cdots h_n) = Concatenation\left(head_1(h_t), head_2(h_t), \cdots, head_n(h_t)\right) W^o \quad (6)$$

where the $head_k(h_t)$ represents the computed attention score by the $k-th$ head, and the W^o denotes the regularization matrix. Every attention head is a linearization mapping matrix. The output of every attention head is computed as a weighted combination of a group of vectors, as presented in equation (7).

$$\{\propto_1, \propto_1, \cdots, \propto_n\} = SoftMax(\frac{QK^T}{\sqrt{d_k}})V \quad (7)$$

$$head_i(h_t) = \sum_{j=1}^{n} \propto_j V_j \quad (8)$$

where the $Q, K,$ and V represent the query, key, and value of the matrix. The q, k_i, v_i are the vectors mapped by the following mapping matrices $W^q, W^k, and\ W^v$, which are learned during training. The mapping is performed with equation (9).

$$q, k_i, v_i = W^q h_t, W^k h_t, W^v h_t \quad (9)$$

Finally, the feed-forward layer is employed for non-linear feature transformation according to equation (10)

$$FeedForward\ (m_t) = max(0, u_t W_1 + b_1)\, W_2 + b_2 \quad (10)$$

Moreover, the residual Add& Norm module is proposed to handle the connectivity between layers without losing information.

$$LayerNorm(In_1 + Ot_1, \cdots In_n + Ot_n) \quad (11)$$

where the $In_i and\ Ot_i$ represent the $i-th$ input element and the $i-th$ output element of the current layer.

C. Distributed Learning in Fog Environment

As depicted in Fig.1, the fog layer consists of two types of nodes. First, the master node is responsible for universal learning. Second, the are multiple worker nodes that perform local learning. Every fog node (master or worker) contains networking, transportation, link, and application module. The identification of intrusions/cyber-attacks within IIoT flow is carried out at the networking and the transportation modules comprising the smart IIoT devices, such as routing and switching devices. The worker nodes of the fog layer manipulate and learn from the traffic data originated from such interconnected IIoT. The proposed forensics framework (see Fig.1) shows that the computational overhead (Deep-IFS training) is similarly distributed on the presented worker nodes of the fog layer to realize distributed learning. This distributed computation overcomes the scalability problems incurred by the centralized cloud computing-based forensics framework. After the Deep-IFS learns, its outputs are forwarded and saved at the master node and employed to detect intrusions/cyber-attacks in newly generated and unidentified IIoT traffic samples. Accordingly, this distributed learning schema enables attaining the optimal parameters for Deep-IFS training and, hence, evades training overfitting.

D. Identifying intrusion in IIoT traffic

The intrusions/cyber-attacks within the IIoT traffic are identified at the master node of the FC layer. After the features are extracted by the transformer network, they are compressed using the pooling layer. Later, the compressed features are fed into full connected layers for computing the final classification decision. For binary classification, the last layer contains two neurons, and the $softmax$ function is used to calculate the probability that traffic records are malicious or normal (legitimate). In a multi-class classification scenario, the final layer neurons are equal to the number of class labels. The probability of each class is also computed using the $softmax$ activation. Since the Deep-IFS has been developed to be deployed in the FC environment, it is trained at every fog node in parallel, and the achieved output of this distributed deployment is transferred to the cloud network. The legitimate IIoT traffic sample is directed to its target (propagated through the network), and the detected intrusions/cyber-attacks are directed to the system of attack mitigation.

IV. Experiments and Analysis

A. Experimental Setup

The overall experiments of this study were performed on an Intel(R) Xeon (R) CPU E5-2670 0@ 2.60GHz (2 processors), with RAM of 256 GB size, under 64-Bit Windows 10 system, and accelerated with NVIDIA-Quadro-k2200. The implementation of the proposed Deep-IFS was carried out using Keras Library and TensorFlow API. For training, the Deep-IFS was trained using Adam optimizer and Batch size of 32 for 100 epochs. The Cooja simulator is employed to simulate the motes of a radio sensor network to establish the IoT edge layer. The CONTIKI-NG framework is employed to implement IoT WSN motes. The fog layer nodes are designed utilizing three laptops, and they are linked with the virtual edge sensors. In the cloud-based platform, the Deep-IFS is deployed in the cloud utilizing an Amazon EC2 virtual server.

B. Evaluation Measures

To assess the performance of the proposed Deep-IFS, evaluation metrics like false-negative (FN), false positive (FP), true negative (TN), true positive (TP) and, were employed for Performance analysis. The value is these metrics are presented as a confusion matrix of the model output. Further, extra evaluation measures such as $Accuracy,\ Fall-out,\ Recall,\ Precision, and\ F1-measure$) and area under the curve (AUC), which are computing according to equations (12), (13), (14), (15), and (16) correspondingly.

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

$$Fall-out\ (F-out) = \frac{FP}{FP + TN} \times 100 \quad (13)$$

$$Precision\ (PRC) = \frac{TP}{TP + FP} \times 100 \quad (14)$$

$$Recall\ (RCL) = \frac{TP}{TP + FN} \times 100 \quad (15)$$

$$F1 - measure\ (F1) = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (16)$$

C. Dataset Description

Bot-IoT dataset: The Bot-IoT data [20] contains the IIoT traffic samples produced by IIoT smart home appliances such as smart thermostats, smart remotely controlled garage, smart-motion-controlled lights, smart fridge-freezers, and smart weather monitoring systems. The data splitting is presented in Table II, where the training set contains 3,037,933 samples of IIoT traffic, and the test set contains 3,668,522 samples of IIoT traffic. Previous studies [18]-[20] often use two groups of features of this data, i.e., group of all features (G1) and a group of best ten features (G2). Hence, we decide to experiment Deep-IFS on both groups. **UNSW-NB15 dataset** [6]: This data consists of nine classes of intrusions, particularly fuzzes, analysis, backdoor, DoS, exploits, generic, reconnaissance, shellcode, and worms. It consists of a total of 42 attributes. The data contain 175341 training samples and 82332 test samples. The distribution of samples across different classes is presented in Table III. This dataset is often used in two groups of features of this data, i.e., group of all features (G1) and a group of best 20 features (G2). In this study, we experiment with the proposed Deep-IFS in binary and multi-class classifications. For binary classification, we use the dataset in two classes (legitimate and attack). For multi-class classification, we use the data with attack categories (i.e., five classes in the *Bot-IoT dataset* and ten classes in the *UNSW-NB15 dataset*).

Table II. The splitting of IoT traffic samples in BoT-IoT Dataset

IIoT Traffic	No.Train Instances	No.Test Instances
Binary class scenario		
Attack	3036915	3668045
legitimate	1018	477
Multi-class scenario		
Theft	119	79
Reconnaissance	115167	91082
DDoS	2027166	1926624
DoS	894463	1650260
legitimate	1018	477

Table III. The splitting of IoT traffic samples in the UNSW-NB15 dataset

IIoT Traffic	No.Train Instances	No.Test Instances
Binary class scenario		
Normal/Legitimate	56000	37000
attack	119341	45332
Multi-class scenario		
Normal	56000	37000
Generic	400000	18871
Exploits	33393	11132
Fuzzers	18184	6062
DoS	12264	4089
Reco.	10491	3496
Analysis	2000	677
Backdoors	1746	583
Shell	1133	378
Worms	130	44

Table IV. The Features of the BoT-IoT Data

Features Used	Feature Names
BoT-IoT (G1)	seq, stddev, N IN Conn P SrcIP, min, state number, mean, N IN Conn P DstIP, drate, srate, max
UNSW-NB15 (G1)	Proto, service, state, spkts, dpkts, sbytes, dbytes, dttl, dloss, sinpkt, djit, swin, tcprtt, smean, dmean, trans_depth, response_body_len, ct_srv_ src, ct_dst_sport_ltm, is_sm_ips_ ports

Table V. The confusion matrix of proposed Deep-IFS in binary classification scenario using the Bot-IoT test set.

	Features Used	Class	Predicted Classes	
			legitimate	**Attack**
Actual Classes	G2	legitimate	460	17
		Attack	8523	3659522
	G1	legitimate	451	26
		Attack	8971	3659074

Table VI. The confusion matrix of proposed Deep-IFS in a multi-class classification scenario using the G1 of the Bot-IoT test set.

		Predicted Classes				
	Traffic	**legitimate**	**DDoS**	**DoS**	**Recon**	**Theft**
Actual Classes	legitimate	462	4	4	6	1
	DDoS	12	1923800	2443	366	3
	DoS	18	2440	1648049	151	2
	Recon	5	14	47	91016	0
	Theft	0	2	1	2	76

Table VII. The confusion matrix of proposed Deep-IFS in a multi-class classification scenario using G2 of the Bot-IoT test set.

		Predicted Classes				
	Traffic	**legitimate**	**DDoS**	**DoS**	**Recon**	**Theft**
Actual Classes	Legitimate	464	3	7	3	0
	DDoS	11	1924585	1832	196	0
	DoS	18	1705	1648478	59	0
	Recon	3	10	20	91049	0
	Theft	0	1	0	1	77

D. Results

This subsection presents the results attained by the proposed Deep-IFS from four experiments on every dataset. The first two experiments train and evaluate the Deep-IFS for classifying attacks from legitimate traffic. The other two experiments train and evaluate the Deep-IFS to discriminate between various categories of traffic data.

BoT-IoT Dataset: For the binary classification, Table V presents the confusion matrix of Deep-IFS. It could be noted that the training the Deep-IFS using the G2 realizes higher TP and TN (460, 3659522) compared to training using G1. This observation indicates better performance. It could be noted that training on either G2 or G1 realizes comparable performance. However, the G2 based Deep-IFS show marginal performance improvements on F1-measure and accuracy. Moreover, two experiments were performed to evaluate the performance of Deep-IFS for the multi-class classification. In the first experiment, G1 is employed for training and evaluating the Deep-IFS and the realized confusion matrix is presented in Table VI. In the second experiment, the Deep-IFS is trained and evaluated using the G2. The corresponding confusion matrix is presented in Table VII, in which most of the misclassifications occur in DoS and DDoS classes. It can also be seen that most confusion happens in DoS and DDoS classes. It can also be seen that the precision of G2 based Deep-IFS outperforms the precision of Deep-IFS trained on G1 with 1% and 8% improvements on both legitimate and theft classes, respectively. Similarly, the recall of G2 based Deep-IFS outperforms the recall of Deep-IFS trained on G1 with 2% on DDoS class. Additionally, it is obvious that the F1-measure attained by G2 based Deep-IFS outperforms the F1-measure of Deep-IFS trained on G1 with 1% and 4% improvements on both legitimate traffic and theft traffic, respectively.

Table IX. Confusion matrix of the Deep-IFS on G1 of UNSW-NB15 test set

Actual classes \ Predicted Class	N	G	E	F	D	R	A	B	S	W
N	36981	4	2	1	3	1	2	4	2	0
G	2	18844	3	4	9	2	4	3	0	0
E	1	3	11111	1	2	6	4	1	3	0
F	2	1	1	6042	1	3	3	5	4	0
D	0	2	4	3	4071	2	5	1	1	0
R	0	1	2	5	2	3481	1	2	1	1
A	1	4	5	1	3	1	656	4	2	0
B	0	1	0	4	4	6	1	564	3	0
S	1	2	1	2	1	8	3	7	353	0
W	0	1	0	0	0	0	0	1	0	42

N: Normal, G: Generic, E: Exploits, F: Fuzzers, R: Reconnaissance, A: Analysis, B: Backdoors, S: Shell, W: Worms.

Table VIII. The confusion matrix of proposed Deep-IFS in binary classification scenario using the UNSW-NB15 *test set.*

	Features Used	Class	Predicted Classes	
			legitimate	**Attack**
Actual Classes	G1	legitimate	36951	49
		Attack	29	45303
	G2	legitimate	36979	21
		Attack	9	45323

Table X. Confusion matrix of the Deep-IFS on the G2 UNSW-NB15 test set

Actual classes \ Predicted Class	N	G	E	F	D	R	A	B	S	W
N	36985	0	2	1	3	1	2	4	2	0
G	1	18856	2	1	4	2	2	2	1	0
E	1	1	11119	1	2	3	0	1	3	0
F	2	1	1	6048	1	3	3	1	2	0
D	0	2	2	2	4077	2	2	1	1	0
R	1	0	2	2	2	3484	1	2	1	1
A	1	2	3	1	2	1	661	4	2	0
B	1	0	0	2	1	3	1	573	2	0
S	1	1	2	2	4	5	3	2	358	0
W	0	0	1	0	0	1	0	0	0	42

N: Normal, G: Generic, E: Exploits, F: Fuzzers, R: Reconnaissance, A: Analysis, B: Backdoors, S: Shellcodem, W: Worms.

Table XI. Comparison between Deep-IFS and the cutting-edge approaches on BoT-IoT

	Binary-class scenario							Multi-class scenario						
	Acc	**PRC**	**RCL**	**F1**	**F-out**	**AUC**	**Training Time (s)**	**Acc**	**PRC**	**RCL**	**F1**	**F-out**	**AUC**	**Training Time (s)**
SVM [20]	99.99	99.99	100	99.99	0	97.1	1557.9	91.31	91.42	90.13	90.77	9.87	89.9	1701.8
RNN [20]	97.91	99.99	97.91	98.94	2.09	98.3	326.2	92.14	93.94	92.84	93.39	7.16	92.4	401.3
LSTM [20]	98.06	99.99	98.06	99.02	1.94	100	415.1	96.1	95.52	94.44	94.98	5.56	97.1	518.7
NV [28]	99.78	99	98	98.50	2	98.7	1408.3	92.04	92.15	91.23	91.69	8.77	91.3	1647.8
Bi-LSTM [30]	98.96	99.99	98.92	99.45	1.08	99.8	689.8	96.67	95.92	96.12	96.02	3.88	97.7	8231.5
CNN [13]	98.45	99.86	98.42	99.13	1.58	98.9	191.4	92.89	97.11	92.84	94.93	7.16	94.8	257.6
VCDL [18]	99.74	99.99	99.75	99.87	0.25	99.1	198.1	93.44	97.52	93.44	95.44	6.56	95.3	247.1
Deep-IFS	**99.75**	**99.99**	**99.75**	**99.87**	**0.25**	**99.9**	**135.6**	**98.1**	**96.95**	**98.0**	**97.5**	**1.9**	**99.7**	**184.8**

UNSW-NB15 dataset: for binary classification, two experiments performed on the G1 and G2 of the data, and the resulting confusion matrices are presented in Table VIII. It could be noted that Deep-IFS show comparable performance on the two data groups with a slight improvement on G2. Further, two additional experiments are carried out on G1 and G2 for multi-class classification and the corresponding confusion matrices are presented in Table IX and Table X. Unlike other DL models, Deep-IFS realized similar performance on both data groups. The observations mentioned above show that our model can extract and distinguish important features in IIoT traffic data, making it more efficient. Discernibly, it also reduces the time required for training and inferencing, as demonstrated in later sections.

E. Comparative Analysis

To validate the importance of the proposed Deep-IFS, we compare its performance against the recently introduced ML and DL approaches presented in Table I. It could be seen that most of the recent studies just addressed the intrusion/anomaly detection in IIoT traffic using predefined or traditional ML or DL approaches. It is also obvious that the security paradigms adopted in these studies were centralized. The distributed security paradigm was addressed in [30], [18]. To ensure fair comparisons, we reproduced the results of the before-mentioned approaches by reimplementing them according to the parameters and configurations reported in their corresponding paper.

Table XI tabularizes the comparative results on the BoT-IoT dataset and also presents the corresponding training time. For the binary-class scenario, the lowest performance was realized by the SVM on all measures. On the other hand, the NB [28] trained with a bijective soft set approach has shown great improvements over the SVM. With regard to recurrent approaches, RNN and LSTM [20] achieved comparable performance. However, the LSTM is more time-consuming due to the gating mechanisms it uses to keep long-term computation [20]. Moreover, in the convolutional approach, the CNN [13] has lower performance than the before-mentioned recurrent networks. Nevertheless, VCDL [18] has revealed great performance on all measures. More importantly, the proposed Deep-IFS achieved a robust performance (Accuracy: 99.77; Precision: 99.99; Recall: 99.77; F1-measure: 99.88). For

, Table XII. Comparison between Deep-IFS and the cutting-edge approaches on UNSW-NB15 dataset

Model	Binary-class scenario							Multi-class scenario						
	Acc	**PRC**	**RCL**	**F1**	**F-out**	**AUC**	**Training time**	**Acc**	**PRC**	**RCL**	**F1**	**F-out**	**AUC**	**Training time**
SVM [20]	81.6	81.91	95.67	88.26	4.33	93.31	457.3	76.12	75.1	76.4	75.74	23.6	83.4	488.3
RNN [20]	97.6	97.14	98.52	97.83	1.48	99.69	91.3	91.15	90.13	93.82	91.94	6.18	93.8	100.2
LSTM [20]	98.82	98.14	98.63	98.38	1.37	99.69	110.4	93.14	92.14	93.11	92.62	6.89	93.5	132.7
NV [28]	80.11	80.22	85.34	82.70	14.66	84.7	408.3	75.18	76.47	78.11	77.28	21.89	82.2	447.1
Bi-LSTM [30]	99.15	98.74	98.9	98.82	1.1	99.71	181.2	97.94	96.35	96.74	96.54	3.26	98.1	210.3
CNN [13]	98.01	97.23	98.21	97.72	1.79	99.69	100.5	85.13	84.11	88.4	86.20	11.6	92.3	115.2
VCDL [18]	98.74	98.9	97.31	98.10	2.69	99.55	98.9	91.25	91.12	90.1	90.61	9.9	95.89	121.4
Deep-IFS	**99.94**	**99.92**	**99.94**	**99.93**	**0.06**	**99.99**	**56.7**	**99.75**	**98.09**	**98.2**	**98.14**	**1.8**	**99.98**	**67.2**

Table XIII. The statistical significance results of Deep-IFS using accuracy measure on both datasets.

Models	**BoT-IoT dataset**	**UNSW-NB15 dataset**
SVM [20]	0.0021	0.0011
RNN [20]	0.021	0.0024
LSTM [20]	0.0187	0.0387
NV [28]	0.0091	0.0027
Bi-LSTM [30]	0.0381	0.0424
CNN [13]	0.0189	0.0054
VCDL [18]	0.0272	0.0097

Table XIV. The ablation study of Deep-IFS.

	BoT-IoT dataset			**UNSW-NB15 dataset**		
	Acc	**F1**	**AUC**	**Acc**	**F1**	**AUC**
Baseline	92.6	91.26	94.42	93.6	92.25	94.43
Local GRU	94.6	95.03	97.82	94.91	93.83	96.71
Local GRU +MHA	95.98	94.35	98.19	97.82	96.38	98.41
Local GRU +MHA+FF	97.01	96.71	98.92	98.24	97.31	98.55
Deep-IFS	**98.10**	**97.5**	**99.7**	**99.75**	**98.14**	**99.98**

the Multi-class scenario. It cab be noted that the performance of all previous models degrades compared with the binary scenario. Nevertheless, the Deep-IFS shows great performance improvement over the other models.

The same models are compared on the UNSW-NB15 dataset and the results are presented in Table XII. For the binary class scenario, it could be noted that the ML models [20],[28] achieve the lowest performance with an accuracy of 81.6% and 80.11%, respectively. The CNN models [13][18] show more improved performance compared with ML models, with an accuracy of 98.45% and 99.74%. Further, the LSTM models [20][30] show higher performance with an accuracy of 98.82% and 99.15% correspondingly. More importantly, the Deep-IFS outperform other models on all measures. In addition, for a multi-class scenario, the performance of all models significantly degrades, whereas SVM and NB achieve accuracy of 76.12% and 75.18%, respectively. CNN [13] and VCDL [18] achieved an accuracy of 85.13% and 91.25% correspondingly. LSTM [20] and Bi-LSTM [30] realized accuracy of 93.14% and 97.94% respectively. The proposed Deep-IFS again achieved robust performance (Accuracy of 99.75, F1-measure of 98.14, and AUC of 99.98) as with binary classification.

The reason that SVM and NB achieve the lowest performance is that they fail to deal with high dimensional data, and they have a poor feature extraction capability. Additionally, the robustness of CNN models over ML is explained by the robust feature extraction capability of convolutional kernels. However, it fails to capture the sequential characteristics of data, which means that the LSTM based model is more efficient for this task. The proposed Deep-IFS combines the advantages of both spatial learning and sequential learning using the local GRU layer and also is able to focus on important features using an attention mechanism.

Furthermore, comparing the training time of the models on BoT-IoT shows that Deep-IFS has the lowest training time with 135.6 and 184.8 seconds for binary and multi-class classification, respectively. Similarly, on the UNSW-NB15 dataset, Deep-IFS also exhibited the lowest training time with 56.7 and 67.2 seconds for binary and multi-class classification. This result further explains the time efficiency of the proposed Deep-IFS.

F. Statistical significance Analysis

To further validate the robust performance of the proposed Deep-IFS, we use a paired t-test to quantify the statistical significance over other models using the accuracy measure. The calculated p-values from the two datasets are presented in Table XIII. These results were calculated in a multi-class classification scenario, which is more challenging than binary classification. It could be seen that the quantified $p - values$ are less than 0.05.

G. Ablation Study Analysis

In our experiments, we select LSTM as the baseline architecture of our model, and we performed an ablation experiment to assess the contribution of different blocks, as presented in Table XIV. It could be noted that implementing the local GRU improves accuracy with 2% and 1.5% on BoT-IoT and UNSW-NB15 datasets, respectively. The MHA layer enhances the accuracy with 1.4% on BoT-IoT and 3% on UNSW-NB15 since it enables the model to focus more on essential representations and ignore useless ones. Accuracy improvement of 1% and 0.5% were achieved by adding the FF layer. The inclusion of the Add & Normalize layer improves the model accuracy with 1% on both datasets as it prevents losing information across the layer and reduces training time.

H. Analysis of Recognition Time

To investigate the impact of distributed learning compared to a centralized learning scheme. Hence, we experiment with the proposed Deep-IFS in both learning schema using the G2 feature data and G1 data and calculated the recognition time incurred in every learning scheme, as presented in Fig.3. It could be seen that the recognition times of the distributed Deep-IFS using trained with the G2 have the lowest recognition time. When trained using the G1, it shows better recognition time than the centralized Deep-IFS. This observation further demonstrates the effectiveness of distributed learning in fog environments compared with centralized learning (i.e., cloud-based learning).

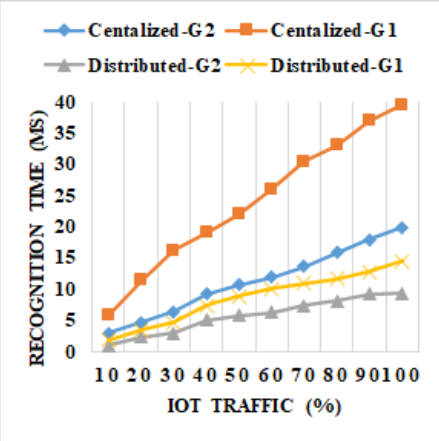

Fig 3. The Recognition time incurred by the Deep-IFS in distributed and centralized learning

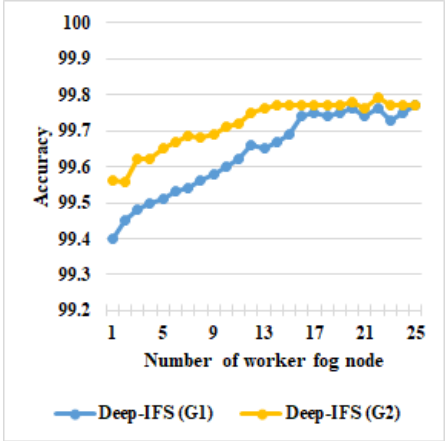

Fig. 4. The impact of the number of worker fog nodes on the accuracy of the proposed Deep-IFS

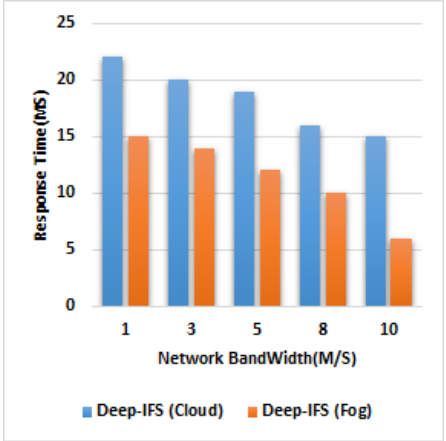

Fig. 5. Average response time for fog-based and cloud- Deep-IFS

I. *Impact of Number of Fog nodes*

To investigate the impact of the number of nodes, we experiment with the proposed Deep-IFS (using the G2 and G) on a varied number of worker fog nodes and compare its accuracy with other models, as depicted in Fig.4. It can be observed that the proposed Deep-IFS exhibits rapid convergence since it attains the best accuracy using 15 nodes, and no accuracy improvement attained beyond that number.

J. *Response Time Analysis*

To assess the proposed deep-IFS efficiency, we compare its implementation in a fog-based and cloud-based IoT platform. In this experiment, the response time is calculated ten times, and the mean value of various network speeds. As presented in Fig. 5, the response time of the fog-based Deep-IFS is less than the cloud-based Deep-IFS as the fog nodes position the computation closer to the edge layer and hence can recognize malware traffic with low latency. The unstructured data that originated from IoT devices could be effectively recognized by Deep-IFS as it can attend to important patterns.

V. Limitations and Future Works

Despite the superiority of the proposed Deep-IFS, it does have some limitations. First, the Deep-IFS is trained in a supervised manner, which prevents learning from unlabeled traffic. Thus, we intend to expand the Deep-IFS to learn from unlabeled traffic using semi-supervised learning i.e., generative networks or self-ensembling. Second, the proposed framework did not address how data privacy will be kept, which is an important aspect of sensitive industrial applications. Thus, we aim to address this challenge using federated learning and privacy-protection techniques in Multi-Access Edge Computing. Further, we also intended to address this limitation using blockchain-enabled fog/edge computing. Third, messaging complexity represents the charge of broadcasting a new chunk to all parties within an IIoT environment, which might worsen the proficiency of alarm aggregation and realizing composite intrusions immediately. Fourth, large volumes of IIoT traffic might lower the efficiency of Deep-IFS in manipulating the incoming traffics with no miss. This could be handled using an ignorance technique that decides the amount of traffic to be neglected when the network traffic surpasses the proposed Deep-IFS supreme processing ability. Furthermore, in plans, we aim to investigate the Deep-IFS interpretability to offer a further transparent framework for sensing the intrusions/cyber-attacks in IIoT traffics. Additionally, we plan to improve this work and apply it in a real-world scenario, especially in smart-manufacturing, smart-transportation, and smart-healthcare applications. Finally, we will investigate our model for large imbalanced data in intrusion detection tasks.

VI. Conclusions

This paper presents a novel DL-approach, called Deep-IFS, for detecting intrusions in IIoT traffic in an FC environment. The main target was to defeat the limited scalability of the current IDS. In order to realize scalable IDS, traffic data samples were distributed across numerous fog workers to learn concurrently from the features of IIoT traffic. Additionally, training and inferencing were achieved by the proposed Deep-IFS employing the LocalGRU layer for local information extraction and using MHSA for global learning. Thus, Deep-IFS helps alleviate the risk of gradient vanishing and parallelize the learning computation (GPU execution), which is not possible in traditional RNNs. The performance of the proposed Deep-IFS approach was assessed with the Bot-IoT dataset. The realized performance of Deep-IFS distributed learning in an FC environment has reduced recognition time compared to centralized learning on the cloud and has realized a substantial performance enhancement compared with cutting-edge DL-based IDS. Moreover, the experimental analysis demonstrated that the Deep-IFS attained better accuracy when trained with G2 than when trained using the entire set of features. Deep-IFS permits simpler communication of data among fog nodes and minimizes overheads providing a useful decision support framework to support the individuals and IIoT service providers to communicate their data in a trusted and secure way.

References

[1] M. M. Hassan, A. Gumaei, S. Huda, and A. Almogren, "Increasing the Trustworthiness in the Industrial IoT Networks Through a Reliable Cyberattack Detection Model," *IEEE Transactions on Industrial Informatics*, vol. 16, pp. 6154-6162, 2020.

[2] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo, and T. Guo, "Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, pp. 6182-6192, 2020.

[3] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines, and J. Coble, "Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data," *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 4362-4369, 2019.

[4] 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 of Things Journal*, 2020.
[5] S. Liaqat, A. Akhunzada, F. S. Shaikh, A. Giannetsos, and M. A. Jan, "SDN orchestration to combat evolving cyber threats in Internet of Medical Things (IoMT)," *Computer Communications*, 2020.
[6] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, 2015, pp. 1-6.
[7] 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 Transactions on Industrial Informatics*, vol. 16, pp. 5244-5253, 2019.
[8] A. N. Jahromi, S. Hashemi, A. Dehghantanha, K.-K. R. Choo, H. Karimipour, D. E. Newton, and R. M. Parizi, "An improved two-hidden-layer extreme learning machine for malware hunting," Computers & Security, p. 101655, 2019.
[9] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys & Tutorials*, 2020.
[10] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," Neurocomputing, vol. 347, pp. 149–176, 2019.
[11] S. Cuomo, V. Di Somma, and F. Piccialli, "A computational method for the European option price in an Internet of Things framework," *Future Generation Computer Systems*, vol. 107, pp. 730-735, 2020.
[12] H. HaddadPajouh, R. Khayami, A. Dehghantanha, K.-K. R. Choo, and R. M. Parizi, "Ai4safe-iot: an ai-powered secure architecture for edge layer of internet of things," Neural Computing and Applications, pp. 1 – 15, 2020.
[13] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.
[14] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, pp. 869-904, 2020.
[15] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: the confluence of edge computing and artificial intelligence," *IEEE Internet of Things Journal*, 2020.
[16] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2020.
[17] N. C. Luong, Y. Jiao, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "A Machine-Learning-Based Auction for Resource Trading in Fog Computing," *IEEE Communications Magazine*, vol. 58, pp. 82-88, 2020.
[18] B. A. NG and S. Selvakumar, "Anomaly detection framework for Internet of things traffic using vector convolutional deep learning approach in fog environment," *Future Generation Computer Systems*, 2020.
[19] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework," *Future Generation Computer Systems*, 2020.
[20] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779-796, 2019.
[21] A. Khan, S. Din, G. Jeon and F. Piccialli, "Lucy with Agents in the Sky: Trustworthiness of cloud storage for Industrial Internet of Things," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2020.2974493.
[22] G. Casolla, S. Cuomo, V. S. d. Cola and F. Piccialli, "Exploring Unsupervised Learning Techniques for the Internet of Things," in IEEE Transactions on Industrial Informatics, vol. 16, no. 4, pp. 2621-2628, April 2020, doi: 10.1109/TII.2019.2941142.
[23] J. Liu, T. Li, P. Xie, S. Du, F. Teng, and X. Yang, "Urban big data fusion based on deep learning: An overview," *Information Fusion*, vol. 53, pp. 123-133, 2020.
[24] N. Misra, Y. Dixit, A. Al-Mallahi, M. S. Bhullar, R. Upadhyay, and A. Martynenko, "IoT, big data and artificial intelligence in agriculture and food industry," *IEEE Internet of Things Journal*, 2020.
[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998-6008.
[26] Dai Z, Yang Z, Yang Y, et al. Transformer-xl: Attentive language models beyond a fixed-length context[J]. arXiv preprint arXiv:1901.02860, 2019.
[27] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, I. Stoica, Spark: Cluster computing with working sets, HotCloud 10 (10-10) (2010) 95.
[28] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, "Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city," *Future Generation Computer Systems*, vol. 107, pp. 433-442, 2020.
[29] F. Piccialli, S. Cuomo, N. Bessis and Y. Yoshimura, "Data Science for the Internet of Things," in IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4342-4346, May 2020, doi: 10.1109/JIOT.2020.2985598.
[30] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A Deep Blockchain Framework-enabled Collaborative Intrusion Detection for Protecting IoT and Cloud Networks," *IEEE Internet of Things Journal*, 2020.

Mohamed Abdel-Basset Received his B.Sc., M.Sc and the Ph.D in operations research from Faculty of Computers and Informatics, Zagazig University, Egypt. He is a head of department of Computer Science, Faculty of Computers and Informatics, Zagazig University. His current research interests are Optimization, Data Mining, Computational Intelligence, Robust Optimization, Engineering Optimization, Multi-objective Optimization, Swarm Intelligence, Evolutionary Algorithms, and Artificial Neural Networks.

Victor Chang is currently a Full Professor of Data Science and Information Systems at the School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough, UK, since September 2019. He co-leads computational Biology and Data Analytics research Group, and is the Research Leader for Beneficial Artificial Intelligence Research Group.

Hossam Hawash Received his B.Sc. from Zagazig University, faculty of computers and informatics, department of computer science, Egypt. His research interests include Optimization, Deep learning algorithms, Swarm Intelligence, Evolutionary Algorithms, and Artificial Neural Networks.

Ripon K. Chakrabortty is a lecturer on System Engineering & Project Management at the School of Engineering and Information Technology, the University of New South Wales (UNSW Australia), Canberra. He obtained his PhD on Computer Science from the same University in 2017, while completed his MSc and BSc from Bangladesh University of Engineering & Technology on Industrial & Production Engineering in 2013 and 2009 respectively. His research interest covers a wide range of topics in Operations Research, Optimization problems, Project Management, Supply Chain Management and information Systems Management.

Michael Ryan is the Director of the Capability Systems Centre, University of New South Wales, Canberra. He lectures and regularly consults in a range of subjects including communications systems, systems engineering, requirements engineering, and project management. He is a co-chair of the Requirements Working Group in the International Council on Systems Engineering (INCOSE). He is a Fellow of Engineers Australia, a Fellow of the International Council on Systems Engineering, and a Fellow of the Institute of Managers and Leaders.