This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3032415, IEEE Transactions on Network Science and Engineering

Deep Learning-enabled Threat Intelligence Scheme in the Internet of Things Networks

Muna Al-Hawawreh, Nour Moustafa, *Senior Member, IEEE*, Sahil Garg, *Member, IEEE,* and M. Shamim Hossain, *Senior Member, IEEE*

***Abstract*—With the prevalence of Internet of Things (IoT) systems, there should be a resilient connection between Space, Air, Ground, and Sea (SAGS) networks to offer automated services to end-users and organizations. However, such networks suffer from serious security and safety issues if IoT systems are not protected efficiently. Threat Intelligence (TI) has become a powerful security technique to understand cyber-attacks using artificial intelligence models that can automatically safeguard SAGS networks. In this paper, we propose a new TI scheme based on deep learning techniques that can discover cyber threats from SAGS networks. The proposed scheme contains three modules: a deep pattern extractor, TI-driven detection and TI-attack type identification technique. The deep pattern extractor module is designed to elicit hidden patterns of IoT networks, and its output used as input to the TI-driven detection. TI-attack type identification is used to identify the attack types of malicious patterns to assist in responding to security incidents. The proposed scheme is evaluated on the two datasets of TON-IoT and N-BAIOT. The experimental results prove that the scheme achieves high performances in terms of the detection and false alarm rates compared with other similar techniques.**

***Index Terms*—Threat Intelligence, deep learning, Internet of Things (IoT), Space, Air, Ground, and Sea (SAGS) networks**

I. Introduction

THE widespread of Internet of Things (IoT) systems should consider the dynamic communications between Space, Air, Ground, and Sea networks (SAGS) to facilitate IoT services to end-users and organizations. IoT systems play a key role in our lives by offering automated services to end-users and organizations. It has created the opportunity for people to integrate and connect physical devices, drones, automobiles, and other embedded applications to the Internet, and remotely control them across cloud systems. IoT technologies have recently offered the full coverage of the entire world by enabling connections in more areas on the earth and space [1]. This has provided a wide variety of benefits, such as improving application visibility, enhancing service performance, reducing economic cost, and enabling better decision making, to society and business [2]. As many organizations have transformed their business to adopt IoT-SAGS technologies, it is estimated that there will be more than 40 billion IoT-connected devices by 2027, with the potential value of the IoT up to $11 trillion by 2025 [3].

Cyberattacks have become serious threats to security and privacy as their impact on IoT-SAGS systems would not cause financial losses, but threaten human safety [4]. Moreover, as IoT-SAGS devices are deployed in the areas of transportation, energy, military, manufacturing, and other paramount ones, attackers can affect the public and national security by breaching sensitive data and compromising critical devices [5], [6]. Examples of companies that have started providing SAGS technologies to facilitate the IoT service in broad-range are Tesla, Google's loon, and TTs Flying COW. Tesla has launched 700 low-cost satellites while Google's loon and TTs Flying COW has employed balloons and Unmanned Aerial Vehicles (UAVs) to provide remote access services [7]. Recent incidents that triggered great concern regarding IoT security were some cyberattacks during the COVID-19 pandemic. For instance, attackers launched phishing and spear-phishing campaigns targeted people who have worked from home and interact remotely with their business infrastructures, in another case, COVID-19 malware targeted medical facilities conducting trials of COVID-19 vaccines to exfiltrate and leak patient information [8].

Recent cyber threats demonstrate the weaknesses of existing cyber defenses, such as firewalls, intrusion detection and prevention systems, as their mechanisms are often built on heuristic and static attacks signatures and cannot detect new variants of attacks [9]. IoT systems are vulnerable to new families of attacks that could exploit attack surfaces of devices and their network protocols. SAGS networks are considerably vulnerable to zero-day, where there is a lack of security considerations for protecting their heterogeneous and complex devices and systems [7], [10]. IoT-SAGS systems require intelligent security systems that can automatically identify evolving cyber threats. Threat intelligence (TI) is a procedure of offering a proactive defense approach that enhances an attack's detection process and reduces its processing time. TI is defined by the National Institute of Standards and Technology (NIST) [11] as ”the aggregation, transformation, analysis, interpretation, and enrichment of threat information to provide the necessary context that can aid decision making”. It is still in its early stage of implementation and faces challenges in

Muna Al-Hawawreh is with the School of Engineering and Information technology, the University of New South Wales @ADFA, Canberra, 2612, Australia (e-mail: m.al-hawawreh@student.adfa.edu.au).

N. Moustafa is with the School of Engineering and Information technology, the University of New South Wales @ADFA, Canberra, 2612, Australia (e-mail: nour.moustafa@unsw.edu.au).

S. Garg is with the Electrical Engineering Department, École de technologie supérieure, Université du Québec, Montréal, QC H3C 1K3, Canada (e-mail: sahil.garg@ieee.org).

M. Shamim Hossain is with the Chair of Pervasive and Mobile Computing, King Saud University, Riyadh 11543, Saudi Arabia, and also with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543,Saudi Arabia. (e-mail: mshossain@ksu.edu.sa).

the IoT industry. This is because it depends on short-term indicators, such as a blacklist IP address, malware hash and malicious URLs, which lacks intelligence related to long-term threat indicators and patterns [9], [12].

Artificial Intelligence (AI)-based TI has attracted significant attention from business and academia due to its effective capability to learn large-scale data and tackle unseen malicious events [13]. More importantly, Deep Learning (DL) can be used to develop adaptive TI models as DL can efficiently deal with unstructured, heterogeneous, and large volumes of IoT-SAGS data [14], [15]. It automatically works without human intervention to extract hidden threat patterns from large-scale data. The research in developing DL-enabled TI models is still considered in its early stage in particular for SAGS networks. Most existing studies have focused on statistical and classical Machine Learning (ML) for building intelligent TI models [16], [17], [18]. However, their models suffered from high complexity, low detection accuracy, and lack of generalization capabilities which made them difficult for a dynamic threat. Other works adopted deep learning techniques with the main focus on specific IoT-SAGS devices' data (e.g., opcodes) which can lead to late threat detection [13], [19], [20], [21]. There is still a research gap of developing an automated AI-enabled TI model for discovering cyber threats in IoT-SAGS networks, that we attempt to address in this study.

We propose a new DL-enabled TI scheme for IoT-SAGS networks with the main focus on utilizing network traffic for detecting attacks. our scheme utilizes deep learning techniques to extract hidden network patterns of cyber attacks to address the challenge of existing TI models (e.g., the lack of generalization). It automatically extracts the appropriate threat knowledge and patterns that can help in understanding and detecting cyber threats, and providing appropriate intelligence that identifies attack types. The main contributions of this work are as follows.

1) An adaptive TI-Deep Pattern Extractor (DPE) module is proposed using a deep sparse auto-encoder algorithm for extracting latent patterns of malicious events.
2) A TI Driven Detection (TIDD) module is developed using a Gated Recurrent Neural Network (GRNN), where the output of DPE is used as input/feeds to the detector engine for identifying abnormal behaviors.
3) A TI-Attack Type Identification (TIATI) module is suggested that identifies attack types using a Deep GRNN (DGRNN) algorithm.

The rest of this paper is structured as follows. Section II explains the background and previous studies of threat intelligence, IoT and SAGS networks. In Section III, the proposed deep learning-enabled threat intelligence model is described. This is followed by the experiments and discussion in Section IV. After this, we conclude the paper in Section V.

II. Background and related work

This section explains the background and related studies of threat intelligence, threat models, SAGS and IoT networks.

A. Overview of Threat Intelligence (TI)

TI is defined as knowledge about threats based on evidence that can help to make decisions [22], emerged to reduce the gap between advanced attacks and defense mechanisms [9]. It is also known as *relevant*, *actionable* and *valuable* information about a cyber threat. The *relevant* aspect includes information about targeted organizations and/or purposes, *actionable* means that the information should be sufficient and specific for performing an action, response or decision, and the *valuable* information must contribute to a beneficial business outcome [23]. TI can be categorized into four main types: *strategic*, *tactical*, *operational* and *technical* [9], [24]. *Strategic* TI includes high-level information related to the financial impact of and decision-making about potential threat risks in an organization and the budget required to mitigate them. The *strategic* TI sources involve local and national media, policy documents from nation-states, and industry and academia produced content (e.g, white paper and research publications). Although the *strategic* TI is useful, it needs a lot of effort to identify the relevant information and valuable insights of cyber threats.

The second category of TI is an *operational* one, which contains information related to specific impending attacks against an organization, and is very rare due to the difficulty of obtaining such private information from an attacker's infrastructure, with governing the only entity capable of gaining details of attacks by accessing attackers' chat forums and other sources (e.g., the darknet). This TI can help in the case of less sophisticated threat groups as they usually discuss their plan in unprotected channels. Obtaining such intelligence is difficult for more sophisticated groups that usually take serious precautions in their discussions and communications. Third, *Tactical* TI is related to an attacker's techniques, which considers tactics and procedures that help in understanding its methodology of defence and employing appropriate policies. This information can be gained from communicating with other peer organizations to know whether they are facing attacks or purchasing for commercial providers. Furthermore, it can be obtained from research articles where researchers provide new tactics and techniques for performing new attacks. In this regard, utilizing encryption techniques with malware to encrypt data is an attack tactic and technique that was initially presented in an academic research paper that inspired the attackers in developing ransomware attacks [25]. Fourth, *Technical* TI is related to the Indicator of Compromise (IoC), which acts as the main source of producing intelligence to feed the investigating and mentoring functions of an organization, such as firewalls, intrusion detection and prevention systems or other appliances, which could include malicious IP addresses, the subject line of phishing emails, payload hashes and other elements. These IoCs are short-shelf life as the attackers keep changing their techniques and procedures; however, other indicators are associated with attacks behavior and patterns can have longer-shelf life.

B. Threat Models in IoT Networks

The best means of securing IoT systems is performing threat modeling as a starting point for examining cyber threats and their motives [26]. It is a structured manner of critical thinking regarding defining the most significant assets of IoT systems, their data flows and trust boundaries, prioritizing potential threats and attacks, and developing appropriate protection and detection countermeasures [27], [28]. Given the layered structure of an IoT architecture, which includes an application, network and edge or perception layers [13], threat modeling can be achieved by decomposing the entire system and observing each layer. The application one represents the IoTśSGAS applications and services with web and mobile clients that track, monitor and control physical SGAS devices at the edge. Most of the threats and attacks associated with this layer are phishing, social media interaction[29], [30] and web application ones [31].

The network layer, which consists of cloud computing, mobile devices and the Internet, transfers information among layers and provides full access to the edge layer using different communication networking such as aerial communications, satellite communication and GPS. Denial of Service (DoS), wormhole and bluejacking are attack examples of this layer [32]. The edge or infrastructure layer includes the most critical SGAS devices that have direct interactions with IoT environments, such as satellites, UAVs, automobiles, users terminal, sensors and edge gateways. The most significant attacks related to this layer are tampering, ransomware, jamming and false data injection [33], [34].

Considering recent research on IoT threat modeling [13], [35], [36], the edge segment is prioritized as the area with the highest risk. This is because most of the value of an IoT-SGAS system resides in this layer which has direct and indirect interactions with other layers and components. Moreover, most of the devices deployed are resource-constrained, with public access, many vulnerabilities and no self-security. As a consequence, this layer is the part most targeted by attackers and its impact can extend to an entire IoT-SGAS system. A recent report [37] stated that more than one million infections have gained access to gateway devices. Therefore, as appropriate protection countermeasures are necessary for this critical segment of an IoT system, it is the focus of this study.

C. SAGS and security challenges

In the current approach of IoT devices communication, most of the remote devices interconnect using traditional earthen networks such as Wi-Fi, Fifth Generation (5G), Worldwide Interoperability for Microwave Access (WiMAX), and Long-Term Evolution (LTE) which offer high-capacity data pipes. Nevertheless, this current communication approach suffers from offering flexible and cost-effective on-demand service and meeting the current requirements of processing and storage among mobile IoT devices. This is extremely difficult for mobile IoT devices in urban areas, where the traffic is dramatically changing in terms of time and space. To handle the challenges, the SAGS network has been emerged as a promising solution to provide IoT devices with cost-effective, low latency, large-scale, reliable, and flexible wireless communications [38]. The SAGS network consists of three network layers including the ground and sea layer, air/aerial layer, and space layer. The ground and sea layer consist of IoT mobile devices such as automobiles, smartphones, smartwatches(i.e., with mobile users), and smart sea ships. The mobile devices are connected using terrestrial communication technologies (e.g., LTE, WIMAX, Wi-Fi, and 5G). The air or aerial layer consists of balloons, and unmanned aerial vehicles (i.e, drones) that can provide high-speed wireless access and the space layer includes satellites [1].

While the development SAGS networks offers great benefits for critical applications, such as military, transpiration and supply chain, there is a risk related to cybersecurity challenges [7]. It is obvious that these challenges come from integrating different and multiple nodes of the SAGS layers, such as roadside infrastructure, automobiles, mobile terminal users, UAVs, sea-ships and satellite nodes, in which each of them has a broad range threat landscape. Like most current systems and networks, cybersecurity is not prioritized and is not taken seriously for protecting network nodes, in particular space nodes (i.e, satellites). These nodes are considered a single point of failure, lack of cybersecurity considerations of TCP/IP communications, involve prolonged life-cycle, and adopt security by obscurity approaches [39], [7]. one of the most noticeable cyber attacks against satellites is the espionage attack, which was performed by the Russia-based cyber-espionage group, named Turla. The Turla group used a ground antenna to detect the IP addresses of satellite users and then used the stolen satellite IP addresses to initiate TCP/IP connections to perform stealthy espionage operations against countries [39].

Another challenge is also associated with other nodes in the aerial layer, including UAB and balloons wireless communications. This stems from their unmanned nature and the in-demand remote access [40]. The maritime/sea and ground infrastructure (e.g. automobiles, smart ships, smart ports, and mobile terminal users) impose a significant challenge due to the human interaction which is considered the weakest link in cyber chain [41], [27]. In general, any potential compromise of these SGAS nodes by recent cyberattacks, such as wannCry or Mirai, can lead to serious and devastating consequences to the human life and national security. The development of intelligent TI has a high business priority, due to its actionable and valuable information that can contribute to providing feeds to existing security mechanisms. TI should be easy to standardize, implement and share [9], [42]. However, existing TI is still a manual investigation of technical threat indicators and intelligence, which results in incomplete, redundant, incorrect and missing useful hidden information. Manual investigations and analyses are usually driven by implementing the Cyber Kill-Chain (CKC) framework designed by Lockheed Martin [43]. Advanced threats against IoT systems typically go through multiple phases, in each of which there is a possibility of extracting intelligence that can help in the early detection and blocking of threats. These phases include reconnaissance, weaponization, delivery, exploitation, installation, command & control, and action on the objective. Extracting TI related to each phase can help in identifying the patterns of potential threats and raising an alarm whenever they exist in IoT networks. Also, it provides more context for these patterns by identifying which CKC phase fits the current threat and can assist decision-makers to make an appropriate response.

Automating TI and implementing AI techniques is becoming of the utmost importance for quickly identifying and evaluating threats that might be missed by a manual investigation, thereby enhancing the visibility of unknown threats and strengthening security [19]. According to a recent study [44], 88% of cybersecurity professionals have adopted AI-enabled solutions, and 91% of them intend to increase the implementation of them. AI-enabled TI using ML has recently gained attention from researchers and businesses for converting collected data into actionable intelligence. However, DL-enabled TI is still a hot topic due to its good capability to deeply analyze hidden patterns, extractions and correlations of threat indicators and intelligence [21], [45]. With the proliferation of IoT-SGAS networks, whereby 'big' data are collected from several distributed components at three layers and the tools and methodologies of attackers are evolving, DL-enabled TI would be the preferred solution for handling these challenges and providing actionable TI [46]. DL coupled with a massive amount of heterogeneous IoT-SGAS network traffic can be extremely valuable for identifying both known and unknown threat patterns, and providing insights and intelligence for protecting such networks and systems.

D. Related work

Several studies for threat hunting and intelligence in Information Technology (IT) and the IoT have been undertaken; for example, HaddadPajouh et al. [13] proposed a security architecture based on a Service-oriented Application (SOA) for protecting the edge layer of an IoT system against known and unknown threats. It was designed using AI-powered modules because of its capabilities to learn from the environment and its suitability for dealing with unknown threats. It consisted of cyber threat hunting, cyber threat attribution and cyber TI modules, with the cyber threat-hunting one used to label an observed behavior as normal or malicious. In the case of a malicious one, the AI-based threat-attribution and intelligence modules were used to find the source of an attack, with the optimum decisions and actions based on the attack campaign recommended. In addition to identifying threats based on the CKC framework, their proposed architecture was evaluated based on service management traits, such as middleware aspects, service types and run times, which demonstrated its superiority over existing ones. Although this architecture worked independently of the endpoints' resources, it was distributed over the three layers of an IoT system which exposed it to networking and security problems that may have affected its performance.

Examples of studies that have presented statistical models for providing TI include that in [16] in which the authors used various statistical information to build a supervised malware intelligence model that could classify any new malware variant in its related family. These statistics included the most frequent basic blocks, which are often in a specific malware family, and the population of each family. Also, the study in [42] introduced a new TI based on a honeypot to collect information related to new attack trends. The extracted attack patterns included the number of occurrences of login attempts, root-trying authentication none, root-failed authentication password, root-trying authentication password, unauthorized login got the remote error, got channel direct TCP/IP request and connection lost. This type of TI would not be very useful as it was not defined as the kind of threat essential in the defensive stage.

Many recent studies have proposed AI-enabled threat hunting and intelligence; for instance, the authors in [21] proposed a model for hunting IoT malware (i.e., an ARM CPU architecture) based on Recurrent NNs (RNNs). It depended on analyzing the frequencies of PoCodes of many malware and benign samples, and showed its better efficiency in terms of accuracy than conventional ML techniques. However, using a small dataset to evaluate the proposed model could yield high estimations of its variations in performance in a real environment. Homayoun et al. [45] proposed a ransomware detection model using a Long Short-term Memory (LSTM) technique and Convolutional NN (CNN) for binary classification and, as the LSTM achieved the best performance, it was used to convert a sequence of application activities into a vector to be fed to a One-class Support Vector Machine (OCSVM) method for performing TI by identifying the particular ransomware family. This model performed well but was highly complex for pre-processing and classifying abnormal observations which could affect the early detection of such types of malware.

Jahromi et al. [47] presented an improved threat-hunting model for IoT malware and ransomware using an ensemble of extreme ML techniques. It achieved a reasonable performance compared with those of standard deep NN models such as the stacked LSTM and CNN. Although the authors argued that it was effective in speeding up the training and detection processes, no testing using performance metrics supported this argument. Similarly, the study in [48] proposed threat-hunting model-based ensemble learning for detecting IoT, Windows and Android malware. Its main idea was to use a separate trained model based on common ML for each feature set which, in turn, transformed to a new feature space. Each trained model was considered a membership function that specified to what degree a pattern was compatible with a specific class. Also, the importance of each model trained on each feature was identified using an assigned weight. The proposed model provided robust performances in terms of its accuracy, F score and detection rate. Typically, as such ensemble models have reduced inter-ability, and is difficult to obtain valuable insights at the end.

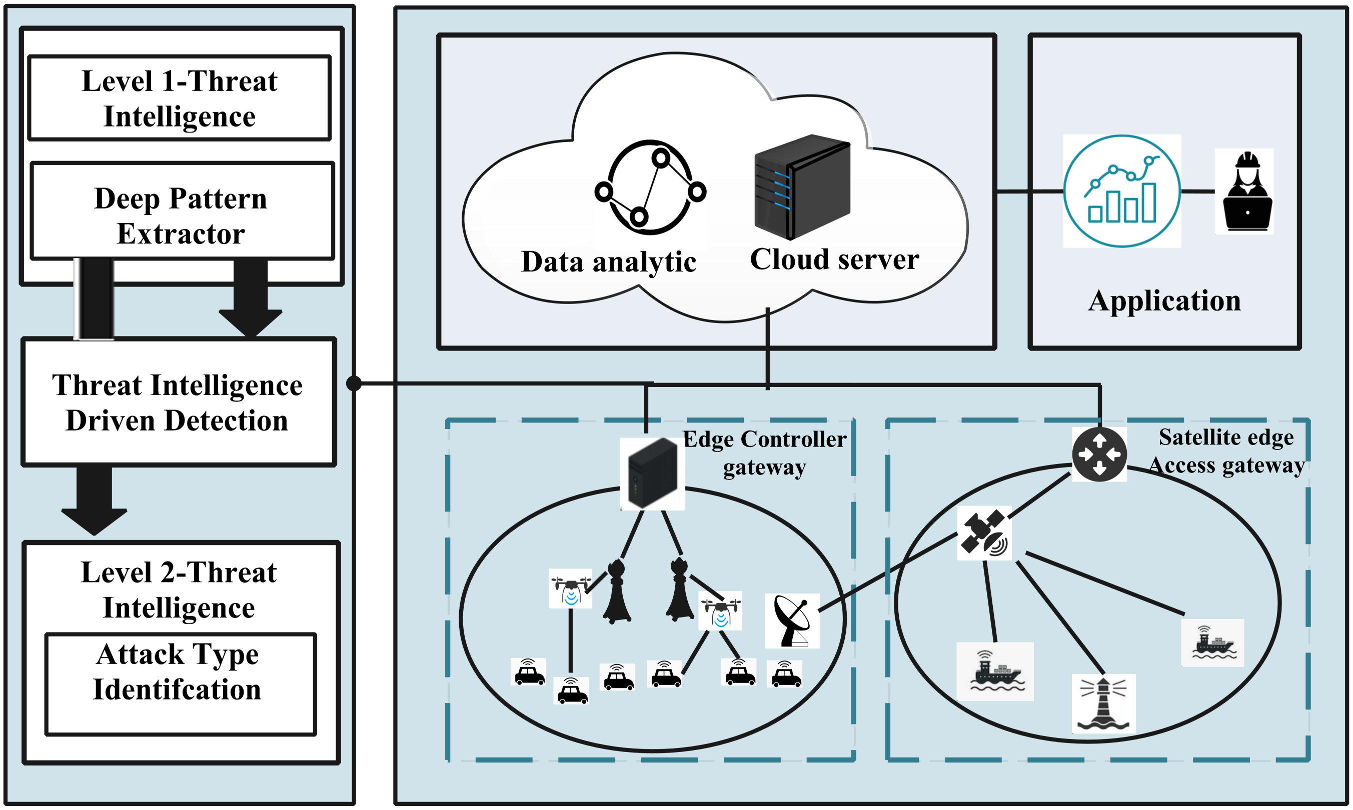

Level 1-Threat Intelligence
Deep Pattern Extractor
Threat Intelligence Driven Detection
Level 2-Threat Intelligence
Attack Type Identifcation
Data analytic
Cloud server
Application
Edge Controller gateway
Satellite edge Access gateway

Fig. 1: Proposed DLTI in SAGS networks.

III. Proposed Deep LEARNING ENABLED THREAT INTELLIGENCE (DLTI) SCHEME

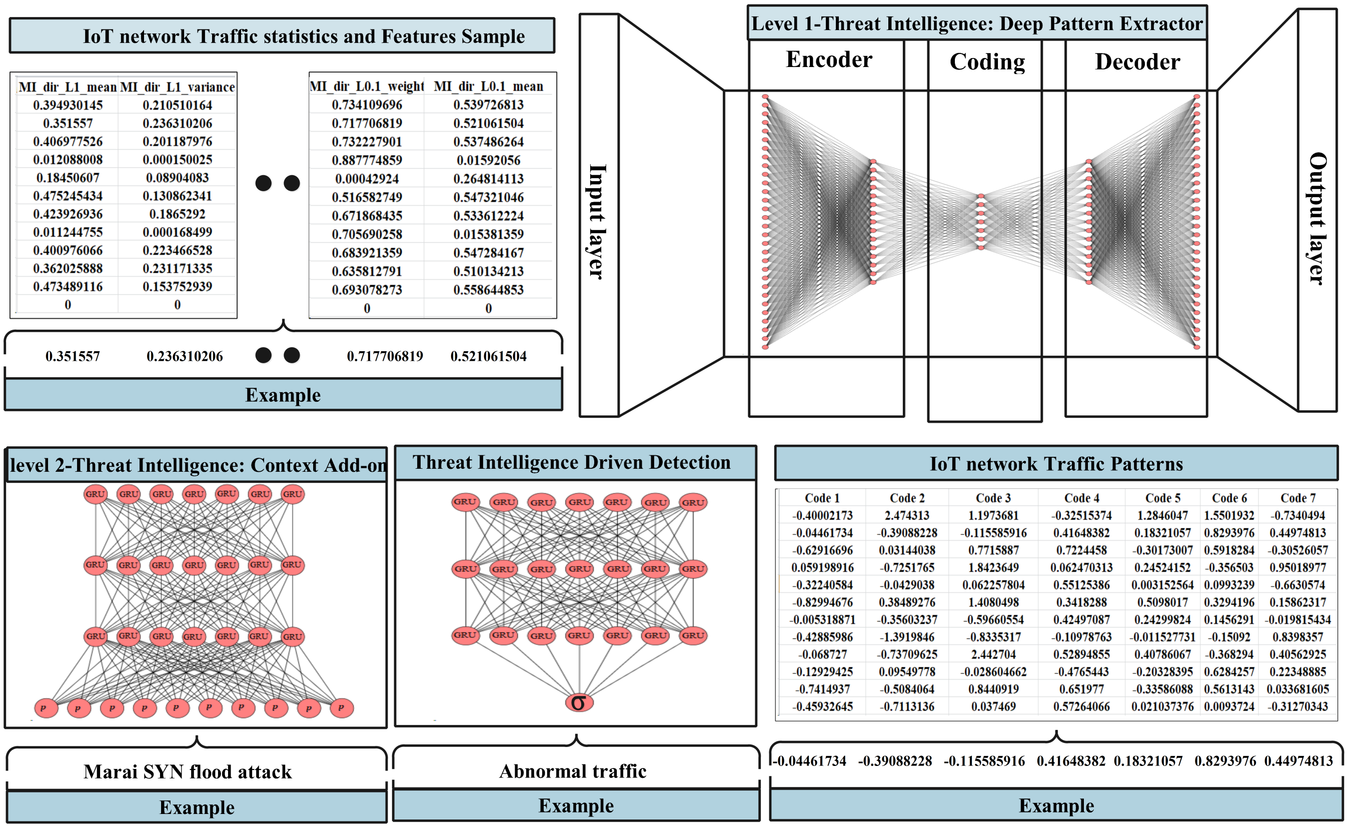
The proposed DLTI scheme is designed based on DL techniques to automatically extract the patterns of meaningful threats as well as detect abnormal behaviors of IoT cloud-edge traffic and identify their types from SAGS networks, as shown in Figure 1. Specifically, network monitoring logs all the traffic passing towards IoT edge-cloud devices analyzes and transforms the captured traffic to observations. Each observation provides valuable data points about the network connections' statistics and features that would help attack detection. However, combining these points to create a pattern is usually performed by a human which leads to many hidden patterns being missed. Therefore, we introduce a DPE module as the first component of our layered DLTI scheme, which represents the first level of TI by exploring knowledge about the network's events and potential threats. As, in the real world, attack traffic is mixed with normal traffic it is extremely difficult to monitor and follow up on big IoT-SAGS traffic data to extract attack or normal patterns. In this regard, the DPE automatically combines network data and creates a new representation with more meaningful and useful network patterns. The abbreviations used in this work are listed in Table I.

TABLE I: Abbreviations and Definitions

Abbreviation	Definition
DL	Deep learning
DPE	Deep Pattern Extractor
TIDD	Threat Intelligence Driven Detection
TIATI	Threat Intelligence Attack Type Identification
DSAE	Deep Stacked Auto-Encoder
NN	Neural Network
GRNN	Gated Recurrent Neural Network

The DPE module is built using a generative deep learning architecture, which has the advantage of learning hidden and unknown patterns without any need to know classes (i.e., attack or normal). It can also make sense of a multitude of data types by extracting general patterns which is extremely useful for analyzing dynamic and heterogeneous IoT-SGAS traffic data and evolving attacks. Also, as this model depends on a black box for defining patterns and coding them in new representations, it solves the privacy issues of using and sharing this intelligence (if required by others).These data extracted by the DPE module are used as feeds to the TIDD technique to determine whether given patterns belong to attacks. Therefore, this technique reduces the reliance on passive forms of attack detection that depend on using traditional intrusion detection models (such as of signatures or rules). It is built based on supervised DL algorithms for identifying abnormal patterns that differ from a normal traffic baseline not previously known based on experience. This leads to reducing the number of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.3032415, IEEE Transactions on Network Science and Engineering


IoT network Traffic statistics and Features Sample
Example
Input layer
Level 1-Threat Intelligence: Deep Pattern Extractor
Encoder
Coding
Decoder
Output layer
level 2-Threat Intelligence: Context Add-on
GRU
P
Marai SYN flood attack
Example
Threat Intelligence Driven Detection
GRU
σ
Abnormal traffic
Example
IoT network Traffic Patterns
Example

MI_dir_L1_mean	MI_dir_L1_variance
0.394930145	0.210510164
0.351557	0.236310206
0.406977526	0.201187976
0.012088008	0.000150025
0.18450607	0.08904083
0.475245434	0.130862341
0.423926936	0.1865292
0.011244755	0.000168499
0.400976066	0.223466528
0.362025888	0.231171335
0.473489116	0.153752939
0	0

●●

MI_dir_L0.1_weight	MI_dir_L0.1_mean
0.734109696	0.539726813
0.717706819	0.521061504
0.732227901	0.537486264
0.887774859	0.01592056
0.00042924	0.264814113
0.516582749	0.547321046
0.671868435	0.533612224
0.705690258	0.015381359
0.683921359	0.547284167
0.635812791	0.510134213
0.693078273	0.558644853
0	0

0.351557 0.236310206 ●● 0.717706819 0.521061504

Code 1	Code 2	Code 3	Code 4	Code 5	Code 6	Code 7
-0.40002173	2.474313	1.1973681	-0.32515374	1.2846047	1.5501932	-0.7340494
-0.04461734	-0.39088228	-0.115585916	0.41648382	0.18321057	0.8293976	0.44974813
-0.62916696	0.03144038	0.7715887	0.7224458	-0.30173007	0.5918284	-0.30526057
0.059198916	-0.7251765	1.8423649	0.062470313	0.24524152	-0.356503	0.95018977
-0.32240584	-0.0429038	0.062257804	0.55125386	0.003152564	0.0993239	-0.6630574
-0.82994676	0.38489276	1.4080498	0.3418288	0.5098017	0.3294196	0.15862317
-0.005318871	-0.35603237	-0.59660554	0.42497087	0.24299824	0.1456291	-0.019815434
-0.42885986	-1.3919846	-0.8335317	-0.10978763	-0.011527731	-0.15092	0.8398357
-0.068727	-0.73709625	2.442704	0.52894855	0.40786067	-0.368294	0.40562925
-0.12929425	0.09549778	-0.028604662	-0.4765443	-0.20328395	0.6284257	0.22348885
-0.7414937	-0.5084064	0.8440919	0.651977	-0.33586088	0.5613143	0.033681605
-0.45932645	-0.7113136	0.037469	0.57264066	0.021037376	0.0093724	-0.31270343

-0.04461734 -0.39088228 -0.115585916 0.41648382 0.18321057 0.8293976 0.44974813

Fig. 2: DLTI structure showing an example of data extraction and TI implementation.

false negative patterns maliciously defined as normal ones. In the second level of TI, ATI module is used to provide more valuable information in order to understand these patterns and identify to which attack they belong so that the appropriate response can be taken by the security team; for example, it can define a specific sequence of patterns that refers to a Marai SYN DDoS botnet, backdoor or other type of attack. This engine is also built on DL techniques and has significant capabilities for generalizing these patterns to various attack types.

A. *TI Level 1- Deep Pattern Extractor (DPE) Module*

A Deep Pattern Extractor (DPE) module automatically extracts new intelligence and patterns from original network data. It extracts the content of collected observations and finds the dependency among features, which transforms into compact and useful pattern representations. A Deep Stacked Auto-Encoder (DSAE) is used to develop the DPE module, as shown in Algorithm 1. It is an unsupervised feed-forward neural network algorithm, with two sub-networks (i.e., encoder and decoder) separated by a code/bottleneck layer. The encoder sub-network, which consists of the input layer and one or more hidden layers, produces the output using only the code/bottleneck layer while the decoder one uses the code layer as input to reproduce the input layer (shown on the left upper side in Figure 2). DSAE works similarly to any type of deep auto-encoder, but restricts the learning process and prevents the copy procedure for input data while adding sparsity to the output of each node in hidden layers, which leads to very few units being activated for each observation of network data. As it optimizes the learning process and generalizes to the unseen data; it can be efficient for extracting meaningful and generalized patterns.

DSAE produces a loss value/reconstruction error, that is $L\left(g_{\grave{\theta}}\left(f_{\theta}\left(x_i\right)\right),\ x_i\right)$, where x_i represents the (i) observation, $f_{\theta}\left(x_i\right)$ represents the output of encoder sub-network and it is calculated based on Equation 1. While the $g_{\grave{\theta}}\left(f_{\theta}\left(x_i\right)\right)$ represents the decoder sub-network output, which is calculated based on Equation 2.

$$f_{\theta}\left(x_i\right) = \sigma\left(w_{x_i} + b\right) \tag{1}$$

$$g_{\grave{\theta}}\left(f_{\theta}\left(x_i\right)\right) = \sigma\left(\grave{w}_{x_i^T} + \grave{b}\right) \tag{2}$$

Here, σ is the desired activation function, θ represents the matrix of weight and bias values [w_{x_i}, b] of the encoder layer, $\grave{\theta}$ the matrix of weights and bias values [$\grave{w}_{x_i^T}, \grave{b}$] of the decoder layer and x_i^T the output from the encoder layer using the coder.

The loss function (L) is a mathematical way of comparing two values ($g_{\grave{\theta}}\left(f_{\theta}\left(x_i\right)\right),\quad x_i$) . In our work, we utilize the Mean Square Error (MSE) as described in Equation 3. DSAE works to predict and reconstruct the input observation rather than classification so the MSE is the best choice for

Algorithm 1: DPE procedures for extracting patterns of IoT network traffic

```
Training-Procedure of DPE (Unlabelled Dataset A)
For each observation in A do
    input-data = get features (new-observation)
    Trained-model = DSAE (input-data)
    Coder = Trained-ModelDecoder Sub-network
End For
Return Coder
Testing-Procedure of DPE (Unlabelled Dataset B, Coder)
code-list = [ ]
For each observation in B do
    input-data = get features (new-observation)
    pattern = Coder (input-data)
    code-list. Add (pattern)
End For
Return new-patterns/codes
```

loss function, and this also has been found from our trial-and-error experiments. Here, n represents the total number of observations during the learning process.

$$L\left(g_{\grave{\theta}}\left(f_{\theta}\left(x_i\right)\right),\ x^i\right)=\frac{1}{n}\sum_i^n\left(g_{\grave{\theta}}\left(f_{\theta}\left(x_i\right)\right)-x^i\right)^2 \tag{3}$$

As the key objective of the learning is to minimize the reconstruction error/loss values $\left(Min\left(L\left(g_{\grave{\theta}}\left(f_{\theta}\left(x_i\right)\right),\ x^i\right)\right)\right)$. The DSAE imposes sparsity constraints on this learning process to optimize the reconstruction process, so the data representation and the meaningful patterns are understood and extracted. This can be achieved by adding activity regularizer function R to the output of each hidden layer as can be described by Equations 4 and 5. Thus, R penalizes the sum of the absolute value of the activation function in the hidden layers (it is called here A_h^i to act as a function for both encoder and decoder hidden layers) for observation number (i), and scales it by sparsity parameter γ.

$$Min\left(L\right)=Min\left(L\left(g_{\grave{\theta}}\left(f_{\theta}\left(x_i\right)\right),\ x^i\right)+R\right)\right) \tag{4}$$

$$R=\gamma\sum_i\left|A_h^i\right| \tag{5}$$

During the training process, the DPE module automatically discovers the content of the network traffic collected, learns its existing patterns and then codes them in a more compact representation in an unsupervised manner; for example, assuming that a collected observation is related to a Marai SYN DDoS botnet attack [49] that infects thousands of IoT devices and its malicious behavior usually has a high packet rate and low packet size (i.e., 74 bytes) [50], the DPE can learn this and generalize it to unseen data (in the case of a similar Mirai flood attack in future). As shown in Figure 2, supposing a network traffic observation has the numerical features' values [0.351557, 0.236310206, $\ldots$, 0.717706819, 0.521061504], the DPE module can learn its hidden patterns and compact them in a new representation (i.e., patterns), which is [-0.04461734, -0.39088228, -0.115585916, 0.41648382, 0.18321057, 0.8293976, 0.44974813].

B. Threat Intelligence Driven Detection (TIDD)

TIDD is a DL technique-based module that identifies malicious behaviors in IoT networks based on the intelligence provided by the DPE module. DL can provide more efficient generalization capabilities than classical ML techniques, which can work efficiently in a case of unseen data. A Gated Recurrent Neural network (GRNN) is used as a base for the detection engines, as presented in Algorithm . In contrast to standard NN, a recurrent one uses the hidden state from the previous timestep (t) in the learning process, that is, $f_{\theta_t}\left(x_t+h_{t-1}\right)$ where x is an input. Each cell of a GRNN [51] consists of two gates, namely, update and reset. The former decides what information should be discarded and what new computed information should be added, and the latter how much computed information from the previously hidden layer should be discarded or ignored. Thus, the GRNN can retain the most useful pattern(s)/code(s) (i.e, the output from the DPE) for detecting abnormal behaviors.

As the DPE module's output is a sequence of patterns, $P=\left(P_{1,}\ P_2,\ldots..P_t\right)$ is carried over the timesteps $t=(1,2,3,\ldots m)$, where m is the number of DPE outputs (patterns/codes). The GRNN accepts a pattern P_t in each timestep t with the previous hidden state h_{t-1} as an input vector. Then, the update gate is calculated by Equation 7, where $w_{p_t}^u$, and w_h^u are the weights of update gate layer for P_t , and h_{t-1} respectively while b is the bias. The activation function σ , which is sigmoid, is used to help the $Gate_{update}$ to decide whether the new computed information is relevant and therefore, to be added to the memory. This can be achieved by transforming these computed values to between 0 and 1.0 0 is not important but 1 is . To control how much computed information of the previous hidden state h_{t-1} is discarded, the $Gate_{reset}$ defined in Equation 8 is used. Based on this value, the current memory content $\acute{h}$ is calculated according to the Equation 9, where the Tanh function is applied for the summation of $w_{p_t}\ P_t$, b and the element-wise between $Gate_{reset}$ and $w_h h_{t-1}$. As a result, all the computed values are regulated and kept within the boundary [-1,1] to prevent some exploding and rendering others insignificant. Similarly, the final content of the memory in the current timestep (t) is calculated by Equation 9. The $Gate_{update}$ is used to determine determine what should be collected from the current memorys content $\acute{h}$ and the previous step h_{t-1} to be passed to the network(i.e, timestep $t+1$).

$$Gate_{update}=\sigma\left(w_{p_t}^u\ P_t+w_h^u\ \ h_{t-1}+b\right) \tag{6}$$

$$Gate_{reset}=\sigma\left(w_{p_t}^r\ P_t+w_h^r\ \ h_{t-1}+b\right) \tag{7}$$

$$\acute{h_t} = Tanh\left(w_{p_t}\ P_t + Gate_{reset} \odot w_h h_{t-1} + b\right) \tag{8}$$

$$h_t = Tanh\left(Gate_{update} \odot h_{t-1} + (1 - Gate_{update}) \odot \acute{h_t}\right) \tag{9}$$

During the training stage, the GRNN repeats the same mathematical processes (Equation 6 to 9) in multiple timesteps based on the number of outputs from the DPE module. Furthermore, to perform the binary classifier task, the output from the GRNN layers is passed to the output layer (with a sigmoid function) to determine the appropriate decision ($\hat{y}$) regarding the sequence of patterns P. This is done by minimizing the loss function value between the actual output (y) and predicted output ($\hat{y}$) for n observations (i.e., batch size) using Equation 10.

$$L\left(y,\ \hat{y}\ \right) = \frac{1}{n}\sum_{i}^{n}\left(\hat{y}^i - y^i\right)^2 \tag{10}$$

This characteristic and the GRNN's way of learning are particularly useful as it is not easy to identify the patterns essential for detecting the abnormal behaviors of IoT-SGAS network traffic (i.e, cloud-to-edge) given the dynamic, large-scale and heterogeneous characteristics of network traffic. The GRNN-based TIDD module learns to retain or ignore particular patterns in each timestep as it sees fit for identifying abnormal/threat behaviors; for example, the patterns of Maria SYN flood attacks extracted from the DPE, i.e., [-0.04461734, -0.39088228, -0.115585916, 0.41648382, 0.18321057, 0.8293976, 0.44974813], are fed to the TIDD module for identification as abnormal behaviors.

Algorithm 2: TIDD procedure for hunting abnormal behavior

```
Procedure TIDD (DPE output DT )
Split $D^T$ into k folds
For each $k^i$ in K folds do
    Set $k^i$ as test set
    Trained-model= Train GRNN ( K-1 )
    Predict = Trained -Model ($k^i$)
     If  Predict == Normal then
        return normal
   else
        return abnormal
End For
End Procedure
```

C. TI Level 2 -Attack Type Identification (TIATI)

Unlike the TIDD module that identifies the abnormal behaviors of IoT-SGAS, network traffic based on patterns extracted by the DPE which does not recognize their exact types of abnormal traffic, the DL technique-based TIATI module described in Algorithm adds a context to the DPE whereby it can recognize to which attack or threat a pattern belongs so that an appropriate response can be taken by a security team. It is built using the GRNN with an output layer that has a softmax function for separating multiple threat types. This GRNN-based TIATI module learns to retain the patterns appropriate for identifying the type of threat while the output layer with a softmax function is used to determine the probability that a particular pattern belongs to a specific type of threat. Suppose that a simple network structure consists of one GRNN layer with t timesteps and the output layer with a softmax function. The input sequence of patterns $P = (P_{1,}\ P_2, \ldots..P_t^m)$ is carried over t timesteps and the network output from the output layer (with a softmax function) is a one-hot encoded C-dimensional vector y. Then, the probability that a one-input P belongs to a specific threat type (y) can be calculated as:

$$p\left(\hat{y}_c = y_c|P\right) = \varrho\left(P\right)_{y_c} = \frac{e^{P_c}}{\sum_j^C e^{P_j}} \qquad (C = 1, 2, \ldots c) \tag{11}$$

To measure the error of the output layer (with a softmax function), the categorical cross-entropy loss, i.e, negative log-likelihood which can be computed over a batch of multiple sequences of size n using Equation 12 is used.

$$L\left(\hat{y}_c, y_c\right) = -\sum_{i=1}^{n}\sum_{c=1}^{C} y_c^{P_i}.log\left(p\left(\hat{y}_{ic} = y_{ic}|P_i\right)\right) \tag{12}$$

Algorithm 3: TIATI procedure for identifying the threat type

```
Procedure CTH (Threat DPE output $D'^T$) )
Split $D'^T$ into k folds
For each $k^i$ in K folds do
    Set $k^i$ as test set
    trained-model= Train GRNN Softmax ( K-1 )
    predict-threat-type = trained-Model ($k^i$)
    decision= predict-threat-type
    return decision
End For
End Procedure
```

A DL-based TIATI engine can identify the threat type and distinguish among several variants which can help a security team perform an appropriate defensive and mitigation procedure; for instance, as can be seen in Figure 2, extracted patterns are recognized by TIATI as a Marai SYN flood attack. Based on this intelligence, a security team can quickly realize that the black Internet Protocol (IP) addresses (i.e., the source of the attack) should be blocked.

IV. EXPERIMENTS AND EVALUATION RESULTS

A. Data Description

To test our scheme, two IoT cloud-edge network datasets from public sources, namely, N-BAIOT [52] and TON-IoT [53], are used. They are chosen because they were built for a TCP/IP communication stack which represents the nature of IoT-SGAS network traffic well(in particular its cloud-edge traffic). The N-BAIOT data include threats Bashlite and Mirai IoT malware, each of which has various threat activities, such as scanning, UDP/TCP flood attacks and spam data, and consists of 13113 normal and 822763 threat observations. The TON-IoT dataset contains threats related to the backdoor, DDoS, DoS, scanning, injection, ransomware, Man-in-the-Middle (MitM), Cross-site Scripting (XSS) and password attacks, and consists of 300000 normal and 161043 threat observations.

B. Data Processing and Evaluation Metrics

Complete observations of the N-BAIOT and TON-IOT datasets are used to train and test the DLTI scheme which depends on DL techniques. Therefore, the collected data must be converted to numerical values and then normalized to prevent bias in the model. Consequently, mapping is used to change each symbolic feature to a number and a min-max scaler to scale the data within a specific range [0, 1]. This scaler maintains the distribution of the original data and does not change its meaningful information.

To evaluate the performance of the proposed scheme, studies of the two datasets use Accuracy (Acc), the Detection Rate (DR), False Positive Rate (FPR), False Negative Rate (FNR), and Mathews Correlation Coefficient MCC. True Positive (TP) and True Negative (TN) are the numbers of malicious and normal observations correctly identified, and False Positive (FP) and False Negative (FN) the numbers of normal and attack observations incorrectly identified. The expressions for calculating these performance metrics are as follows:

1) The Accuracy is the total number of observations correctly identified, that is,

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

2) The DR is the total number of attack observations correctly identified, that is,

$$DR = \frac{TP}{TP + FN} \tag{14}$$

3) The FPR is the number of normal observations incorrectly identified as an attack, that is.

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

4) The FNR is the number of attack observations incorrectly identified as normal, that is,

$$FNR = \frac{FN}{FN + TP} \tag{16}$$

5) The MCC is used to define the quality of the model and how it performs for identifying the IoT-SGAS network traffic. It has a value range of -1 to 1, with 1 indicating a perfect IoT malicious network traffic detector and -1 an always imperfect one, that is,

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{\left(\begin{array}{c} (TP + FP) * (TP + FN) * \\ (TN + FP) * (TN + FN) \end{array} \right)}} \tag{17}$$

Our DLTI model is developed using Python language on Windows 10 with 16 GB RAM and an i7-8550U CPU processor. For the experiments on each dataset, a random sample of 70% of its total size is used to train the DPE and 30% for testing. For the TIDD, TI-ATI and other existing models, 10-k cross-validations are used for training and testing to guarantee that each observation is evaluated at least once. The results of averaging these cross-validations are also used to correctly adapt and tune the DLTIs parameters (based on trial-and-error experiments), as described in Table III.

TABLE II: DLTIS SCHEME PARAMETERS

DPE	Number of hidden layer neurons (30,15,7,15,30), activaction Function (hidden-selu, output-sigmoid) , lossfunction (Mean Square Error), optimizer (RMSprop), batch size (250), epoch (200)
TIDD	Number of hidden layer neurons (30,30, 30,1), activation Function (hidden-Tanh, recurrent and output-sigmoid), loss function (binary crossentropy), optimizer (RMSprop), batch size (250), epoch (30)
TI-ATI	Number of hidden layer neurons (30,30, 30, ToN-IOT 10, and N-BAIOT 11), activation Function(hidden-Tanh, recurrent-sigmoid and output-Softmax), loss function (categorical crossentropy), optimizer (RMSprop), batch size (250), epoch (30)

C. Result Evaluations

The performance of the proposed DLTI scheme is evaluated on the N-BAIOT and TON-IoT datasets in terms of their DR, Acc, FPR, FNR and MCC. The first part of our proposed scheme, that is, DPE-TIDD, is compared with four techniques, namely, the K-nearest Neighbor (KNN), Nave Bayes (NB) and Logistic Regression (LR), to validate its effectiveness for revealing malicious behaviors in the IoT-SAGS network, as shown in Figure 3, 4, 5, and 6. It can be observed that the performances of the DLTI scheme are better than those of the other techniques using the two datasets. As shown in Figure 3, and 4, for the N-BAIOT one, our proposed DPE-TIDD obtains the best results of a 100% DR, 100% Acc and 99.8% MCC, with the lowest FPR and FNR of 0.00%. The averages of the performances of the other techniques are 99.84-99.98% Acc, 99.98% DR and 0.37-9.15% FPR.

As shown in Figure 5 and 6, for the TON-IoT dataset, our proposed DPE-TIDD obtains the best performance in terms of its quality, that is, MCC (98.73%) and achieves the highest Acc (99.85%) and lowest FPR (00.60%). Although the NB performs best in terms of the DR, that is, 99.62%, and the lowest FNR of 0.38%, it has the poorest MCC (57.17%) which means that it cannot work well with heterogeneous IoT-SAGS network traffic. NB assumes the independency between features and this makes it fails in detecting more advanced attacks that have long patterns (combined more features). The other techniques average a 84.35-88.09% Acc, 91.94 DR, 13.98-22.8 % FPR, 80.06% FNR and 69.87-75.51% MCC. It is worth mentioning that our scheme obtains these results using only 7 input data dimensions for both datasets whereas the other techniques adopt all the features or data points of the network traffic (40 for the TON-IOT and 113 for the N-BAIOT datasets).

Furthermore, to demonstrate the robustness and reliability of DPE-TIDD performance, we use Receiver Operating Characteristics (ROC) curve for both datasets. RoC curve is one of the most important model's performance metrics as it defines the degree of separation between normal and attack observations. In Figure 7 and 8, the ROC curves describe the performance of DPE-TIDD for the two datasets in terms of TPR and FPR with the area under the green and blue lines is a measure of DPE-TIDD accuracy. It can be seen from both figures that the green and blue curves are near the highest point in the upper left corner which represents the near-optimal or perfect performance where all positive and negative observations are correctly identified. Furthermore, the DPE-TIDD is reliable and robust as it performs consistently well for both datasets.

Table III and IV shows the performances of the DPE and TIATI engines when working together in our proposed scheme to add context to the threat patterns/intelligence extracted by recognizing the types of malicious IoT network traffic patterns in both datasets compared with those of the other models, the KNN, LR and NB, in terms of their DRs of attack types (the numbers of observations of specific types correctly defined). For the N-BAIoT dataset, the DPE-TIATI can identify the attack types of malicious IoT network-SAGS traffic belonging to botnets such as combo, junk, scan and TCP flood, achieving averages of 76.09-99.84%. For Marai botnet activities, such as scan, UDP and UDP plain flood attacks, the DPE-TIATI obtains averages of 99.50-99.98%. Other techniques, such as the NB and KNN, perform worse for identifying attack types, in particular bash, combo, junk and Marai scan, with averages of 17.74-79.69%. The KNN and LR achieve the worst identification performances for the bash TCP flood attack, with averages of 2.05-2.20%. For other attacks, such as bash scan, bashUDP flood, Marai Ack, SYN, UDP and UDP plain flood, they obtain averages of 52.39-100%. It can be seen that the NB achieves the best DR for the bashlite UDP flood and Mirai SYN attacks while the LR is the best for identifying a Mirai ACK one.

For the ToN-IoT network dataset, the DPE and TI-ATI can recognize backdoor, DoS, DDoS, MitM, password, ransomware, XSS, scanning and injection patterns, with averages of 99-99.99%. The other techniques achieve similar performances despite the fact that our proposed scheme adopts only 7 input features instead of all those in the dataset. The results obtained by the proposed DLTI and other mechanisms imply the formers superiority for extracting the hidden patterns of IoT-SAGS network attacks that help the detection of normal and abnormal malicious observations. The key reasons behind its performance are as follows. Firstly, using the DPE as the

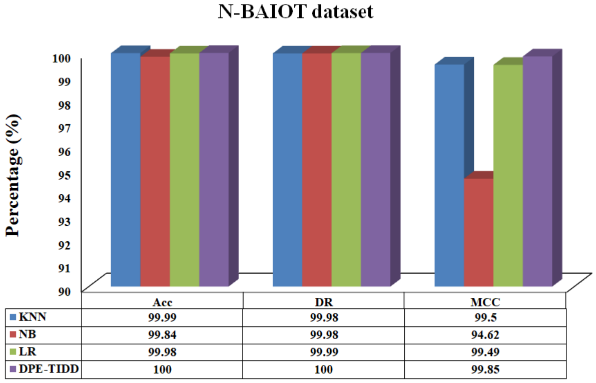


Fig. 3: Performance metrics for N-BAIOT dataset.

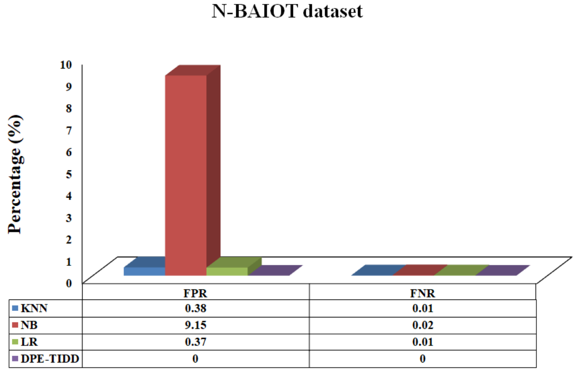


Fig. 4: False positive and negative rates for N-BAIOT dataset.

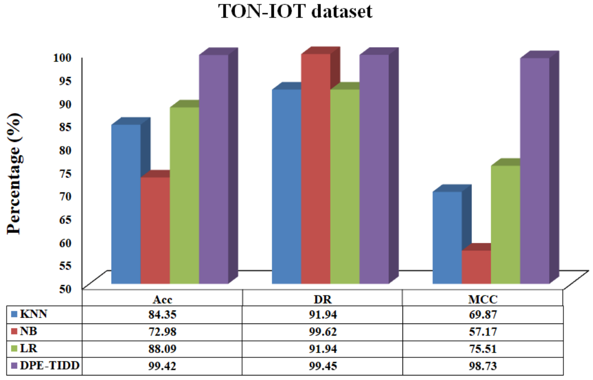


Fig. 5: Performance metrics for TON-IOT dataset.

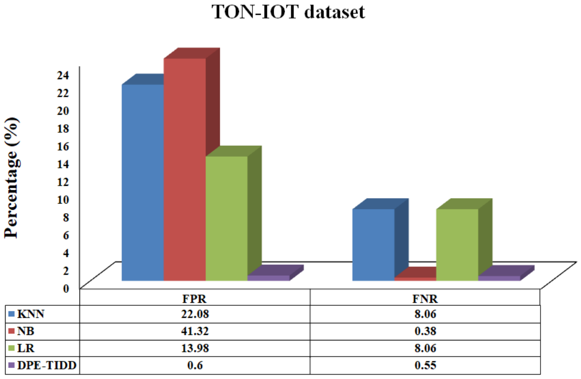


Fig. 6: False positive and negative rates for TON-IOT dataset.

TABLE III: DPE AND TIATI ENGINES ATTACK TYPE DETECTION RATE FOR TON-IOT DATASET

Model	Backdoor	DoS	MitM	Password	Ransomware	Scanning	XSS	Injection	DDoS
KNN	100	100	98.19	99.01	98.05	99.77	96.15	99.22	100
NB	100	99.99	99.90	100	100	100	100	100	99.99
LR	100	100	98.18	99.04	98.05	99.77	96.16	99.22	100
DPE-TIATI	99.99	99.79	99.89	99.85	99.61	99.98	99.96	99.97	99.91

TABLE IV: DPE-TIATI ENGINES ATTACK TYPE DETECTION RATE FOR N-BAIOT DATASET

Model	Bash-combo	Bash-junk	Bash-scan	Bash-TCP-flood	Bash-UDP-flood	Mirai-ACK	Mirai-scan	Mirai-SYN	Mirai-UDP	Mirai-UDP-plain
KNN	79.69	63.93	97.48	2.05	52.39	99.99	57.12	99.85	99.73	69.87
NB	68.00	69.44	98.59	69.12	84.12	98.64	17.74	100	98.00	98.75
LR	79.70	63.94	97.52	2.20	54.4	99.99	99.91	99.74	99.85	99.73
DPE-TIATI	99.55	99.84	99.84	76.09	52.45	99.90	99.50	99.98	99.92	99.98

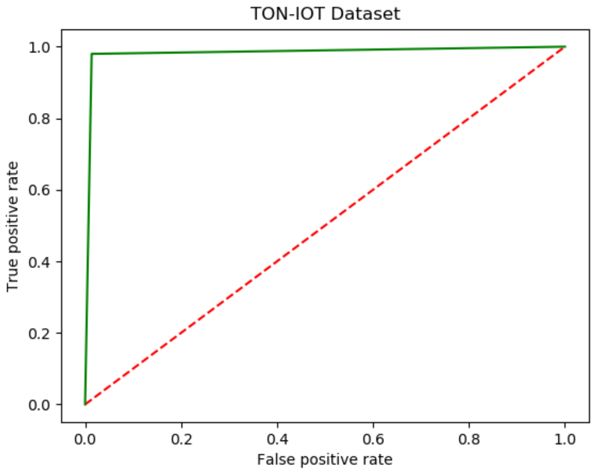


Fig. 7: ROC curve for TON-IOT dataset

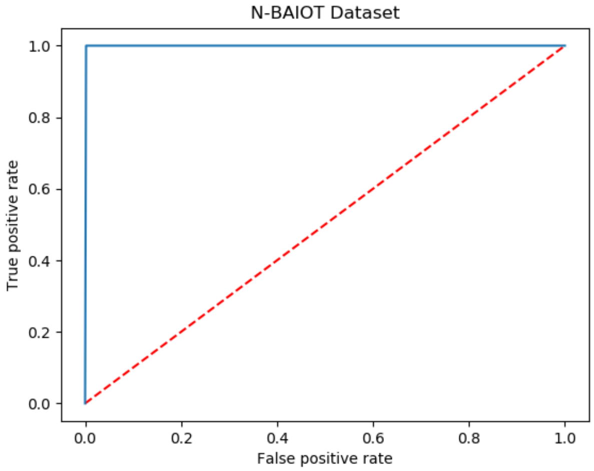


Fig. 8: ROC curve for N-BAIOT dataset.

first level of TI for extracting deeper meaningful network patterns and feeding them to suffix engines, building the DPE using the DSAE technique which has the capabilities to discover the hidden and unknown patterns in an unsupervised manner and make sense of many types of network traffic observations (i.e., different attack types), and transforming/coding them into a new compressed representation.

It is more efficient in dealing with large amounts of heterogeneous IoT network traffic. Secondly, using the output from the DPE as feeds to enrich the detection mechanism, that is, the TIDD, can improve detection quality and reduce reliance on traditional intrusion detection models that depend greatly on static rules and signatures. The TIDD is built using a GRNN which has more capabilities than other deep and classical learning techniques to combine relevant patterns as it sees fit for a specific type of behavior (i.e, normal or attack). It can also retain relevant information and extracted knowledge for a long time while ignoring irrelevant data. Finally, our scheme adopts the ATI engine as the second level of TI that can identify the type of threat/attack which will help a security team choose the procedure for an appropriate response. Developing this engine based on a GRNN with a softmax function in the output layer provides a good capability to retain and combine the significant patterns of each attack type.

D. Time Complexity for our proposed DLTI Scheme

We analyze the complexity of our proposed DLTI scheme by measuring how long a deep learning algorithm will take to produce the final output or result. Practically, this can be calculated using Big ”O” Notation which describes how well an algorithm scales and the time complexity. As explained in Section III, the DLTI scheme consists of three modules that are powered by deep learning techniques: DPE-based on DSAE, TIDD, and TIATI-based on GRNN. The time complexity is separately for each model can be calculated based on O(n) where n is the number of edges in the network. For the DPE module, in the simple standard architecture of SAE, there is an Input layer (I), Hidden layer(H), and Output layer (L) for mapping observation features or network data points from input to output. We assess the weighted sum of inputs from the previous layer to the next in the forward step (i.e., input to hidden (IH) and hidden to output (HL) while we compute the error and then spreads it through the network to update the weights in the backward pass. Thus, the time complexity for standard SAE with *m* number of training observations and *e* is epochs can be calculated as follows *O(n) = (e*m*(IH + HL))*. As described in Table I, we have 5 hidden layers (30,15,7,15,30) with 200 epochs and 322731, 585114 training data observations for TON-IoT and N-BAIoT datasets respectively, the final *O(n)= 200*m*(I*30+30*15+15*7+7*15+15*30+ 30*L)*.

In the standard architecture of GRNN network for TIDD and TIATI modules, there are input layer (I), a recurrent GRNN layer (number of cells in each block) (H), and an output layer (L) for mapping observations or network data points from the whole history of previous inputs to each output. Thus, the time complexity can be calculate as follows $O(n)= 2IH+2H^2+H+HL$. In our implementation, we have input 7 units which are the output of DPE, 30 cell units for each GRNN layer and therefore the O(n) value equals *O(n)= 3*7*(30)+2*900+2*30+30*L* and this gets multiplied by 3 (number of GRNN layers), 30 epochs, the total number of observations (m) and 10 (for K-cross validation). Overall, as we use a small input (this is the advantage of using the DPE module), the final complexity of the GRNN network for TIDD and TIATI is dominated by H(H+L)factor. A Large number of output units(i.e., number of classes) and GRNN cells in each layer can make the complexity of learning a bit expensive.

The computational complexity of other models i.e, KNN, NB, and LR can be described by the number of observations *m*, and the number of features *I* (similar to the input layer of deep learning). Thus, the *O(n)= K*m*I* for KNN (K= number of neighbors), *O(n)= m*I* for NB, and *O(n)= m*I*for LR. Although the time complexity for such models is less than it for DL models, DL is better in terms of its performance and capabilities for the IoT-SAGS network. This is because the DL-based model or scheme's performance is continuously improved with increasing the size of training data as it learns the data distribution and representation efficiently. Whilst other traditional models' performance can reach the state of no change as these models cannot learn the deep representation for input data. This makes DL-based models or schemes much preferable choice for IoT-SAGS networks with particular emerging quantum computing for addressing the high DL resources requirements.

E. Discussions

Our DLTI scheme has many advantages that enable the dynamic behaviors of IoT-SAGS networks and evolving attack techniques to be monitored. First of all, our scheme is scalable, lightweight, and flexible as it is designed based on three modules (i.e., DPE, TIDD, and TIATI) which can be deployed separately in the IoT-SAGS edge network. This way of dividing the entire defense mechanism into small modules where each module is dedicated to a specific function highly fits the complex and large scale structure of the IoT-SAGA network. They can be easily deployed in the production environment, modified and evolved without any effect on the entire system. These traits make our scheme much better than traditional models where the defense mechanism or TI-solution is deployed as a single system and all functions are performed in one block. Such a traditional system or model design brings complexity for deployment, maintenance, and evolution of defense mechanisms in IoT-SAGS networks. It also leads to poor scalability and overload in computing resources which limits the capabilities for securing IoT-SAGS network.

Our modules are not fully decoupling as we have the first level of TI, that is, the DPE module is used to extract traffic patterns to pass them as feeds to the other modules (i.e, TIDD and TIATI). This in its return provides a little dependency and a cost in the communication among them but this limitation can be exposed by providing this level of TI as optional for other modules. This means other modules TIDD and TIATI can work without DPE and using deep learning techniques in powering them gives them this advantage. Another advantage of our scheme is its ability in dealing with high-volume, heterogeneous, and dynamic IoT-SAGS network traffic and the hidden patterns within it. Also, as it overcomes reliance on traditional detection mechanisms by using extracted TI as feeds for the detection process, it can efficiently detect known and unknown attack patterns. Moreover, as it can professionally specify an attack-type, an appropriate response can be made.

On the other hand, our scheme highly depends on deep learning techniques in powering its three modules (i.e., DPE, TIDD, and TIATI) which provide high capabilities to deal with heterogeneous and dynamic traffic of the IoT-SAGS network and the evolving techniques and tactics of attackers. Our scheme uses the DSAE to power the DPE module because of its way of learning from data in an unsupervised way and add sparsity on the hidden layers' output which prevents copying the input observation and force the network on learning the hidden pattern and representing it in fewer dimension. Furthermore, it utilizes artificial NN with recurrent connections for long-term memory (i.e., GRNN) which is better than other types of deep learning methods in terms of training time. It takes a shorter time and remembers the data with long patter which makes it a good choice for a dynamic system and the advanced threats with long patterns. Also, it has less output error which means it has better generalization capabilities compared with other deep learning techniques. However, choosing the appropriate network parameters is a non-trivial process that requires many empirical experiments and it takes long processing time. Moreover, a DL technique deals with only numerical values, a problem we solve by using transformation as a pre-processing step.

V. Conclusion and future work

In this paper, a new scheme, called DLTI, has been proposed for extracting meaningful cyber threat patterns from IoT cloud-edge traffic of Space, Air, Ground, and Sea (SAGS) networks that can aid in detecting attacks. This proposed scheme consists of a DSPAE-DPE engine for automatically learning the hidden and unknown patterns of IoT traffic without requiring knowledge of what is being looked for. It codes and represents these discovered patterns in new forms that can be used as feeds to a second engine called the DGRNNTIDD for identifying abnormal IoT network traffic based on its prior experience. The second level of TI is provided by the ATI engine built based on the GRNN-output layer with a softmax function to add context to the extracted patterns by identifying their malicious types. The proposed DLTI model can capably extract threat patterns from heterogeneous and dynamic IoT network traffic using the ToN-IoT and N-BAIOT datasets. Its performance using the extracted patterns as feeds to the TIDD engine proves the good quality of these patterns and helps the model define abnormal traffic. Also, the second level of TI (ATI) demonstrates a reasonable performance for identifying malicious pattern types. In future work, we plan to evaluate our schemes performance using a real IoT system and investigate obtaining TI from IoT devices, such as their logs. Furthermore, we plan to utilize a microservice architecture to develop and evolve our DLTI scheme.

Acknowledgment

This project has been funded by the Spitfire Memorial Defence Grant, Project number PS39150. The authors are also grateful to the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia for funding this work through the Vice Deanship of Scientific Research Chairs: Chair of Pervasive and Mobile Computing.

References

[1] N. Kato, Z. M. Fadlullah, F. Tang, B. Mao, S. Tani, A. Okamura, and J. Liu, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 140–147, 2019.

[2] P. Brous, M. Janssen, and P. Herder, "The dual effects of the internet of things (iot): A systematic review of the benefits and risks of iot adoption by organizations," *International Journal of Information Management*, vol. 51, p. 101952, 2020.

[3] M. S. Jalali, J. P. Kaiser, M. Siegel, and S. Madnick, "The internet of things promises new benefits and risks: A systematic analysis of adoption dynamics of iot products," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 39–48, 2019.

[4] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for iot: Toward universal and resilient systems," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018.

[5] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of iot new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1606–1616, 2018.

[6] Y. Qian, M. Chen, J. Chen, M. S. Hossain, and A. Alamri, "Secure enforcement in cognitive internet of vehicles," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1242–1250, 2018.

[7] C. Zhao, M. Shi, M. Huang, and X. Du, "Authentication scheme based on hashchain for space-air-ground integrated network," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[8] M. S. Hossain, G. Muhammad, and N. Guizani, "Explainable AI and mass surveillance system-based healthcare framework to combat covid-i9 like pandemics," *IEEE Network*, vol. 34, no. 4, pp. 126–132, 2020.

[9] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Computers & security*, vol. 72, pp. 212–233, 2018.

[10] N. Moustafa, G. Misra, and J. Slay, "Generalized outlier gaussian mixture technique based on automated association features for simulating and detecting web application attacks," *IEEE Transactions on Sustainable Computing*, 2018.

[11] C. Johnson, M. Badger, D. Waltermire, J. Snyder, and C. Skorupka, "Guide to cyber threat information sharing," National Institute of Standards and Technology, Tech. Rep., 2016.

[12] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[13] 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.

[14] M. Al-Hawawreh and E. Sitnikova, "Industrial internet of things based ransomware detection using stacked variational neural network," in *Proceedings of the 3rd International Conference on Big Data and Internet of Things*, Melbourne, Australia.

[15] X. Yang, T. Zhang, C. Xu, S. Yan, M. S. Hossain, and A. Ghoneim, "Deep relative attributes," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1832–1842, 2016.

[16] V. Ghanaei, C. S. Iliopoulos, and R. E. Overill, "Statistical approach towards malware classification and detection," in *2016 SAI Computing Conference (SAI)*. London, United Kingdom: IEEE, 2016, pp. 1093–1099.

[17] N. Khurana, S. Mittal, A. Piplai, and A. Joshi, "Preventing poisoning attacks on ai based threat intelligence systems," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, Espoo, Finland.

[18] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, "Know abnormal, find evil: frequent pattern mining for ransomware threat hunting and intelligence," *IEEE transactions on emerging topics in computing*, vol. 8, 2017.

[19] S. Mittal, A. Joshi, and T. Finin, "Cyber-all-intel: An ai for security related threat intelligence," *arXiv preprint arXiv:1905.02895*, 2019.

[20] M. Al-Qurishi, M. Al-Rakhami, A. Alamri, M. Alrubaian, S. M. M. Rahman, and M. S. Hossain, "Sybil defense techniques in online social networks: A survey," *IEEE Access*, vol. 5, pp. 1200–1219, 2017.

[21] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," *Future Generation Computer Systems*, vol. 85, pp. 88–96, 2018.

[22] R. McMillan, "Definition: threat intelligence," Gartner. www.Gartner.com, Tech. Rep., 2013.

[23] H. Almohannadi, I. Awan, J. Al Hamar, A. Cullen, J. P. Disso, and L. Armitage, "Cyber threat intelligence from honeypot data using elasticsearch," in *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*. Cracow, Poland: IEEE, 2018, pp. 900–906.

[24] A. Chuvakin and A. Barros, "How to collect, refine, utilize and create threat intelligence," Retrieved 10/9/2014 from Gartner. http://www.gartner. com/document/2738618, Tech. Rep., 2016.

[25] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," *Computers & Security*, vol. 74, pp. 144–166, 2018.

[26] S.-J. Yun and J. Kim, "A study on security requirments analysis through security threat modeling of home iot appliance," *The Journal of Society for e-Business Studies*, vol. 24, no. 2, pp. 113–124, 2019.

[27] A. A. A. El-Latif, B. Abd-El-Atty, M. S. Hossain, S. Elmougy, and A. Ghoneim, "Secure quantum steganography protocol for fog cloud internet of things," *IEEE Access*, vol. 6, pp. 10 332–10 340, 2018.

[28] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Transactions on Sustainable Computing*, 2019.

[29] X. Yang, T. Zhang, C. Xu, and M. S. Hossain, "Automatic visual concept learning for social event understanding," *IEEE Transactions on Multimedia*, vol. 17, no. 3, pp. 346–358, 2015.

[30] S. Qian, T. Zhang, C. Xu, and M. S. Hossain, "Social event classification via boosted multimodal supervised latent dirichlet allocation," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 11, no. 2, 2015.

[31] S. N. Swamy, D. Jadhav, and N. Kulkarni, "Security threats in the application layer in iot applications," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2017, pp. 477–480.

[32] R. Sharma, N. Pandey, and S. K. Khatri, "Analysis of iot security at network layer," in *2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE, 2017, pp. 585–590.

[33] J. Zhang, R. S. Blum, and H. V. Poor, "Approaches to secure inference in the internet of things: Performance bounds, algorithms, and effective attacks on iot sensor networks," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 50–63, 2018.

[34] M. Al-Hawawreh, F. den Hartog, and E. Sitnikova, "Targeted ransomware: A new cyber threat to edge system of brownfield industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7137–7151, 2019.

[35] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Secure data sharing and searching at the edge of cloud-assisted internet of things," *IEEE Cloud Computing*, vol. 4, no. 1, pp. 34–42, 2017.

[36] T. Varshney, N. Sharma, I. Kaushik, and B. Bhushan, "Architectural model of security threats & theircountermeasures in iot," in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE, 2019, pp. 424–429.

[37] A. O. Kupreev, E. Badovskaya, and A. Gutnikov, "Ddos attacks in q4 2018," *Kaspersky Lab Report*, 2019.

[38] M. S. AL-Hawawreh and A. I. Zreikat, "Performance analysis of a wimax network in different propagation models," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 15, no. 1, 2017.

[39] G. Falco, "Cybersecurity principles for space systems," *Journal of Aerospace Information Systems*, vol. 16, no. 2, pp. 61–70, 2019.

[40] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on uav cellular communications: Practical aspects, standardization advancements, regulation, and security challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3417–3442, 2019.

[41] S. Lagouvardou, "Maritime cyber security: concepts, problems and models," *Kongens Lyngby, Copenhagen*, 2018.

[42] E. M. Hutchins, M. J. Cloppert, R. M. Amin *et al.*, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, no. 1, p. 80, 2011.

[43] Webroot, "Game changers: Ai and machine learning in cyber security," Tech. Rep., 2017.

[44] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, R. Khayami, K.-K. R. Choo, and D. E. Newton, "Drthis: Deep ransomware threat hunting and intelligence system at the fog layer," *Future Generation Computer Systems*, vol. 90, pp. 94–104, 2019.

[45] S. Kumar, B. Janet, and R. Eswari, "Multi platform honeypot for generation of cyber threat intelligence," in *2019 IEEE 9th International Conference on Advanced Computing (IACC)*. IEEE, 2019, pp. 25–29.

[46] 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*, vol. 89, p. 101655, 2020.

[47] H. Darabian, A. Dehghantanha, S. Hashemi, M. Taheri, A. Azmoodeh, S. Homayoun, K.-K. R. Choo, and R. M. Parizi, "A multiview learning method for malware threat hunting: windows, iot and android as case studies," *World Wide Web*, vol. 23, no. 2, pp. 1241–1260, 2020.

[48] M. Al-Hawawreh, E. Sitnikova, and F. den Hartog, "An efficient intrusion detection model for edge system in brownfield industrial internet of things," in *Proceedings of the 3rd International Conference on Big Data and Internet of Things*, Melbourne, Australia, 2019, pp. 83–87.

[49] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A new threat intelligence scheme for safeguarding industry 4.0 systems," *IEEE Access*, vol. 6, pp. 32 910–32 924, 2018.

[50] M. Al-Hawawreh and E. Sitnikova, "Leveraging deep learning models for ransomware detection in the industrial internet of things environment," in *2019 Military Communications and Information Systems Conference (MilCIS)*. Canberra, Australia: IEEE, 2019, pp. 1–6.

[51] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *International conference on machine learning*, 2015, pp. 2067–2075.

[52] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiotnetwork-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[53] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *IEEE Access*, pp. 1–1, 2020.

Muna Al-Hawawreh received the B.E and M.E in computer science, and is now working towards a Ph.D degree from the University of New South Wales (UNSW), Canberra, Australia. She works as a research assistant at UNSW Canberra Cyber. In her PhD research, Muna developed the world's first ransomware framework targeting IIoT edge gateway in the critical infrastructure. 1n 2019, she was awarded the first prize for high impact publication in the school of Engineering and Information technology (SEIT), UNSW Canberra. She is a reviewer of high-impact factor journals such as the IEEE Internet of Things Journal, IEEE Access, and IEEE Transactions on Dependable and Secure Computing. She is also a program committee member and reviewer for the number of cybersecurity conferences. Her research interests include Software Engineering, Wireless Sensor Network, Cloud, Industrial Control Systems, Internet of Things, Cybersecurity, and Deep and machine learning. Her current work addresses cyber security issues in the industrial internet of things by designing detection models for sophisticated threats.

Nour Moustafa (M'15–SM'19) is a Lecturer at SEIT, University of New South Wales (UNSW)'s UNSW Canberra Australia, and Helwan University, Egypt. He was a Post-doctoral Fellow at UNSW Canberra from June 2017 till December 2018. He received his PhD degree in the field of Cyber Security from UNSW Canberra in 2017. He obtained his Bachelor and Master degree of Computer Science in 2009 and 2014, respectively, from the Faculty of Computer and Information, Helwan University, Egypt. His areas of interest include Cyber Security, in particular, Network Security, IoT security, intrusion detection systems, statistics, Deep learning and machine learning techniques. He has several research grants with totalling over AUD 1.2 Million. He has been awarded the 2020 prestigious Australian Spitfire Memorial Defence Fellowship award. He is also a Senior IEEE Member, ACM Distinguished Speaker, as well as CSCRC and Spitfire Fellow. He has served his academic community, as the guest associate editor of IEEE transactions journals, including IEEE Transactions on Industrial Informatics, IEEE IoT Journal, as well as the journals of IEEE Access, Future Internet and Information Security Journal: A Global Perspective. He has also served over seven conferences in leadership roles, involving vice-chair, session chair, Technical Program Committee (TPC) member and proceedings chair, including 2020IEEE TrustCom and 2020 33nd Australasian Joint Conference on Artificial Intelligence.

Sahil Garg (S'15, M'18) is a postdoctoral research fellow at cole de technologie suprieure, Universit du Qubec, Montral, Canada. He received his Ph.D. degree from the Thapar Institute of Engineering and Technology, Patiala, India, in 2018. He has many research contributions in the area of machine learning, big data analytics, security and privacy, the Internet of Things, and cloud computing. He has over 50 publications in high ranked journals and conferences, including 30+ IEEE transactions/journal papers. He received the IEEE ICC best paper award in 2018 in Kansas City, Missouri. He serves as the Managing Editor of Springer's Human-Centric Computing and Information Sciences journal. He is also an Associate Editor of IEEE Network, IEEE System Journal, Elsevier's Applied Soft Computing, Future Generation Computer Systems, and Wiley's International Journal of Communication Systems. In addition, he also serves as a Workshops and Symposia Officer of the IEEE ComSoc Emerging Technology Initiative on Aerial Communications. He has guest-edited a number of Special Issues in top-cited journals, including IEEE T-ITS, IEEE TII, the IEEE IoT Journal, IEEE Network, and Future Generation Computer Systems. He serves/served as the Workshop Chair/Publicity Co-Chair for several IEEE/ACM conferences, including IEEE INFOCOM, IEEE GLOBECOM, IEEE ICC, ACM MobiCom, and more. He is a member of ACM.

M. Shamim Hossain (SM'09) is a Professor at the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He is also an adjunct professor at the School of Electrical Engineering and Computer Science, University of Ottawa, Canada. He received his Ph.D. in Electrical and Computer Engineering from the University of Ottawa, Canada in 2009. His research interests include cloud networking, smart environment (smart city, smart health), AI, deep learning, edge computing, Internet of Things (IoT), multimedia for health care, and multimedia big data. He has authored and coauthored more than 285 publications including refereed journals conference papers, books, and book chapters. Recently, he co-edited a book on Connected Health in Smart Cities, published by Springer. He has served as cochair, general chair, workshop chair, publication chair, and TPC for over 20 IEEE and ACM conferences and workshops. He is the chair of IEEE Special Interest Group on Artificial Intelligence (AI) for Health with IEEE ComSoc eHealth Technical Committee. Currently, he is the cochair of the 3rd IEEE ICME workshop on Multimedia Services and Tools for smart-health (MUST-SH 2020). He is a recipient of a number of awards, including the Best Conference Paper Award and the 2016 ACM Transactions on Multimedia Computing, Communications and Applications (TOMM) Nicolas D. Georganas Best Paper Award, and the 2019 King Saud University Scientific Excellence Award (Research Quality). He is on the editorial board of the IEEE Transactions on Multimedia, IEEE Multimedia, IEEE Network, IEEE Wireless Communications, IEEE Access, Journal of Network and Computer Applications (Elsevier), and International Journal of Multimedia Tools and Applications (Springer). He also presently serves as a lead guest editor of IEEE Network, ACM Transactions on Internet Technology, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) and Multimedia systems Journal. Previously, he served as a guest editor of IEEE Communications Magazine, IEEE Network, IEEE Transactions on Information Technology in Biomedicine (currently JBHI), IEEE Transactions on Cloud Computing, Future Generation Computer Systems (Elsevier). He is a senior member of both the IEEE, and ACM.