Hindawi
Security and Communication Networks
Volume 2021, Article ID 8873195, 13 pages
https://doi.org/10.1155/2021/8873195

Research Article

Deep Learning-Based Security Behaviour Analysis in IoT Environments: A Survey

Yawei Yue,[1,2] Shancang Li,[2] Phil Legg,[2] and Fuzhong Li[1]

[1]*School of Software, Shanxi Agricultural University, Jinzhong 030801, Shanxi, China*
[2]*Department of Computer Science and Creative Technologies, UWE Bristol, BS16 1QY, UK*

Correspondence should be addressed to Shancang Li; shancang.li@uwe.ac.uk

Received 30 August 2020; Revised 22 November 2020; Accepted 15 December 2020; Published 8 January 2021

Academic Editor: Honghao Gao

Internet of Things (IoT) applications have been used in a wide variety of domains ranging from smart home, healthcare, smart energy, and Industrial 4.0. While IoT brings a number of benefits including convenience and efficiency, it also introduces a number of emerging threats. The number of IoT devices that may be connected, along with the ad hoc nature of such systems, often exacerbates the situation. Security and privacy have emerged as significant challenges for managing IoT. Recent work has demonstrated that deep learning algorithms are very efficient for conducting security analysis of IoT systems and have many advantages compared with the other methods. This paper aims to provide a thorough survey related to deep learning applications in IoT for security and privacy concerns. Our primary focus is on deep learning enhanced IoT security. First, from the view of system architecture and the methodologies used, we investigate applications of deep learning in IoT security. Second, from the security perspective of IoT systems, we analyse the suitability of deep learning to improve security. Finally, we evaluate the performance of deep learning in IoT system security.

1. Introduction

The advancement of network theory and architecture in line with the development of sensors and microprocessors paved the way for the Internet of Things, and applications such as smart homes and smart cities are now becoming widely adopted. According to Gartner, 5.8 billion endpoints will be deployed in 2020, a 21% increase from 2019 [1]. The market for IoT was valued at $190 billion in 2018 and is projected to reach $1102.6 billion by 2026, exhibiting a compound annual growth rate (CAGR) of 24.7% in the forecast period [2]. Banking and financial services have the greatest market share, followed by information technology and telecommunications. Healthcare and government applications also account for a large proportion of the total IoT market. The explosive growth of IoT offers the potential for billions of devices to be connected and exchanging data for various applications. The unique characteristics that IoT offers have also brought a series of new security and privacy threats, of which are a major concern for sustainable growth of IoT adoption.

Often, IoT devices are reported to have vulnerabilities due to their limited resources which can make them an attractive target for attack. With billions of devices interconnected, many and other connected devices launched a targeted attack at the domain name provider Dyn [3], causing a denial of service (DoS) attack against many popular websites such as GitHub, Twitter, and others. Many of the devices used for this attack by the Mirai botnet were using default usernames and passwords. Connected autonomous vehicles (CAVs) are a unique form of IoT, yet attacks have been demonstrated to show how an Internet-enabled vehicle could be controlled remotely through a vulnerability in the media control system that could cause serious physical harm [4]. To be efficient and lightweight to deploy, many IoT applications run on embedded CPUs with limited memory and battery capacity. Many IoT system designs highlight the limitation in computing efficiency as a potential attack vector for security and privacy concerns. IoT devices are widely used as core controllers in critical infrastructures, and they convey valuable information. Stuxnet

[5] is a well-documented malicious computer worm that targeted a specific industrial control system (Uranium Enrichment Plant), which suspended the progress of nuclear weapons program of Iran. IoT technologies play a crucial role in enhancing real-life applications, such as healthcare, smart home, and surveillance.

Given the complexity of developing IoT systems integration, this can potentially provide a wide attack surface for an adversary. Like the Mirai botnet, devices that have weak authentication requirements can be easily compromised and controlled as part of an attack; as the number of connected devices increases, this attack surface continues to grow.

In this paper, we study how deep learning can be used to enhance security and privacy in the IoT era. Firstly, we review security and privacy concerns in IoT systems. We then survey deep learning-based IoT security and privacy applications and develop a taxonomy to consider these works from the viewpoint of deep learning algorithms used and the IoT security problems that they solve. Finally, we present the future research trends and challenges that we have identified. The main contributions of this paper are summarised as follows:

(1) We summarize and provide a taxonomy of recent work using deep learning to enhance the security and privacy property of IoT system and how deep learning can help to build a secure IoT system

(2) We identify the weaknesses that still exist in current research and the discrepancies between these weaknesses and the requirements of the IoT setting

(3) We investigate the possible future research directions toward deep learning enhanced IoT security

2. Background

Minerva et al. introduced the architecture of an IoT system and highlighted the set of features that a system must possess in order to be considered as an IoT system [6]. The main features include the following:

(1) *Interconnection of Things*. Here, the "Thing" means smart object that can collect, create, process, and store data from a user or application perspective.

(2) *Connectivity*. The IoT provides Internet connectivity for objects in the system, including devices, applications, and key IoT infrastructures.

(3) *Uniquely Identifiable Things*. IoT devices should be uniquely identifiable.

(4) *Ubiquity*. The IoT system can provide services that are available for users anywhere and at any time.

(5) *Sensing/Actuation Capability*. As the key component senses the environment, a smart sensor can collect data from environment and transmit this to the IoT systems. An actuator can conduct specific operations depending on the commands received from the IoT system.

(6) *Embedded Intelligence*. Advances in artificial intelligence are to be embedded into edge IoT systems.

(7) *Interoperable Communication Capability*. An IoT system should be able to communicate using standard and interoperable communication protocols.

(8) *Self-Configurability*. Due to the fact that there are a large number of heterogeneously connected devices in an IoT system, it is natural that IoT devices may need to manage and configure themselves, which could range from software and hardware management to resource allocation.

(9) *Programmability or Software Defined*. Physical devices in IoT systems can be easily customized with a user's command or software defined functions without physical changes.

In our previous works, we defined a service-oriented architecture (SoA) for the general IoT [7], as shown in Figure 1. This paper extends previous works by detailing the sensing layer, network layer, service layer, and interface layer. The sensing layer is integrated with available hardware objects to sense the statuses of things. The network layer is the infrastructure to support wireless or wired connections among things. The core of this architecture is the service layer, which consists of service discovery, service composition, service management, and service interfaces. The service layer allows developers to meet the request of end users with minimal workload. The interface layer consists of the interaction methods with users or applications. We adopt this architecture for the remainder of the paper.

3. Behaviour Modelling and Analysis of IoT Using Deep Learning

Deep learning (DL) is considered to be the founding pillar of modern artificial intelligence [8]. DL has been widely used in computer vision, speech recognition, robotics, and many other application areas. Compared with traditional machine learning techniques, deep learning has some key advantages. (1) The use of many hidden layers within a neural network structure means that deep learning can fit complex nonlinear relationships between attributes. (2) Popular architectures such as convolution neural networks (CNNs) and long short-term memory (LSTM) networks have the ability to extract and identify useful features directly from raw data (e.g., autoencoders) instead of relying on hand-crafted statistical features as performed in traditional machine learning. (3) Deep learning is particularly well suited for dealing with 'big data' challenges [9].

With billions of devices interconnected together to sense and share information worldwide, IoT systems naturally produce a huge volume of data. Deep learning has significant potential to help analyse user (events, apps) behaviours in complicated IoT systems. Furthermore, deep learning could enable IoT devices to learn complex behaviour patterns more effectively than traditional learning techniques.

The IoT is a complete ecosystem that contains a variety of devices and connections, a tremendous number of users, and a huge volume of data. To identify the potential vulnerabilities that exist within an IoT system, it is necessary to look at the whole IoT ecosystem and the behaviours exhibited

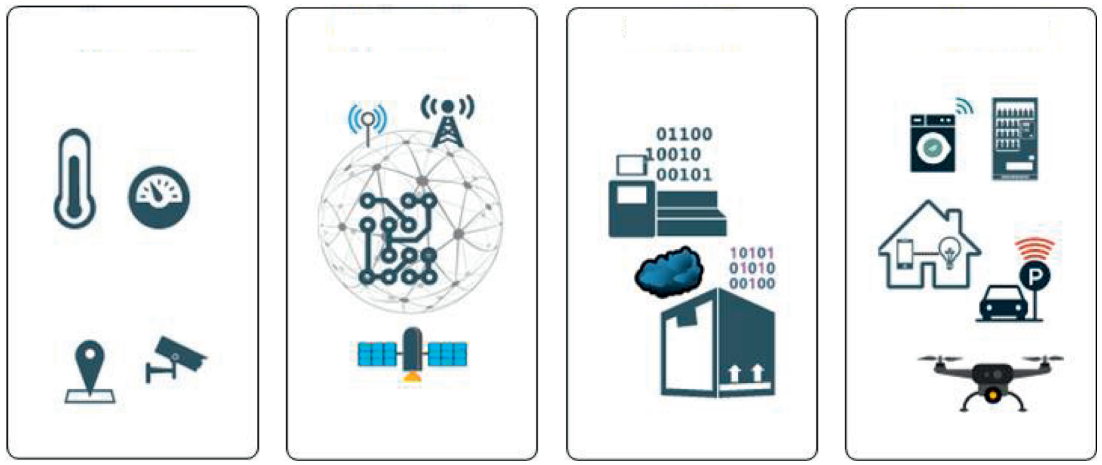

Figure 1: Service-oriented architecture (SOA) of the IoT system: (a) sensor layer; (b) network layer; (c) service layer; (d) interface layer.

rather than to focus on the individual device or layers. In this work, we focus on the following three problems: (1) to identify the uniqueness of each IoT device by classifying, training, and extracting device fingerprint of an IoT device; (2) to investigate the network behaviours in IoT; and (3) to model data abuse in IoT environment.

3.1. Identify the Uniqueness of IoT Devices Using DL. Each device in an IoT system will often have some fixed features, such as physical characteristics or services that it provides. Based on such features, we can profile a device to uniquely identify it from other IoT devices in the same system.

For example, an IoT digital camera could be used to take photographs and record audio/video and could even link with social networking data sources if permitted access. The CCD sensor in the digital camera has a unique sensor pattern noise (SPN) which could be used to create a unique fingerprint of the device. Such fingerprints for IoT devices could also be identified based on the device users, which can be further analysed using techniques such as deep learning. Having a means to fingerprint an IoT device and specifically the data generated by the IoT device (rather than merely relying on serial numbers, IMEI numbers, and so on) would be particularly beneficial should there be a need to identify malicious usage of devices in a complex interconnected IoT system. Likewise, this notion of fingerprinting can also be used as part of authentication and trust between connected devices:

(1) *Device Identification Using DL.* Traditional methods of device identification may use serial numbers, IMEI codes, or other static identifiers; however, these can potentially be spoofed or manipulated by an attacker. Deep learning has the potential to identify subtle differences between classes when considering a large feature set to characterise data and therefore could be effective for device identification as discussed previously. Deep learning methods can extract features from the signal or traffic produced by the device in order to recognise and identify the device.

An example of this is camera model identification, where the objective is to determine the device that captured the image provided. Work in [10] proposes the method of using deep CNNs to automatically extract features to identify the capture device. They calculate residual noise in the image by subtracting a denoised version of the image from the image provided. The residual noise is then used as input to the CNN model to extract and identify distinct features from various device types. Work in [11] uses a CNN to extract model-related features and then uses a support vector machine (SVM) to predict the camera model. In both of these cases, the role of deep learning is primary for the feature extractor. Similar examples have also been applied for audio device identification [12].

Radio fingerprinting has also been studied where devices are identified by their wireless radio device properties. In [13], Yu et al. propose a solution using partially stacking-based convolutional DAE to classify devices through reconstructing a high-SNR signal. Based on RF fingerprinting techniques, Bassey et al. proposed a framework to detect unverified smart devices with deep learning [14]. First, they use a convolutional neural network to automatically extract high-level features from RF traces; then, they perform dimensionality reduction and decorrelation on deep features. Finally, they use clustering techniques to classify IoT devices.

(2) *Service Fingerprint Extraction Using DL.* Due to the dynamic nature of IoT networks, it can be difficult to maintain static fingerprints for devices as they are connected or removed from the network. Therefore, establishing a dynamic behaviour baseline is essential. Fingerprinting IoT devices can also be a challenge due to the heterogeneous nature of IoT devices, protocols, and command interfaces. Service

fingerprints identify IoT devices based on the services that they provide, which then generates a profile that can be used to identify the type of device that it is likely to be. Typically, this would be achieved using system logs and web traffic as inputs to extract behavioural fingerprints.

Previously, researchers have employed machine learning to address challenges in IoT [15–17]. Meidan et al. propose an IoT device classification framework based on HTTP packet analysis [15]. They perform this as a two-pass classification to firstly distinguish between IoT devices and non-IoT devices and then perform a fine-grain classification model to differentiate between nine distinct IoT devices. In [16], the authors propose to approximately model IoT behaviour by the collection of communication protocols used, and the set of request and response traffic sequences observed, from which device features are then extracted from the network traffic. Finally, features are aggregated using a statistical model as a base profile for device identification. In [17], the proposed scheme extracts up to 23 features from each packet, from which they form a fingerprint matrix and use a random forest to develop a classification model.

More recently, deep learning has been adopted for IoT behaviour fingerprinting. Reference [18] proposes to use information from network packets to identify devices. They observed that packet interarrival time (IAT) is unique among devices. They extract and plot the IAT graph for packets where each graph contains 100 IATs. Then, they use the CNN to learn features from device graphs and distinguish different devices. Another study in [19] attempts to automatically identify the semantic type of a device by analysing its network traffic. First, they define a collection of discriminating features from raw traffic flows, and those features are used to characterise the attributes of devices.

Then, they use a LSTM-CNN model to infer the semantic type of a device. Due to the large variety of devices and manufacturers in IoT setting, other researchers [20] argue that traditional intrusion detection methods cannot suitably detect compromised IoT devices given the scale of devices being monitored. They propose DI¨OT, a self-learning distributed anomaly-based intrusion detection system, to identity compromised devices. DI¨OT can effectively build device-type-specific behaviour profile with minimal human efforts. Federated learning is utilized in DI¨OT to efficiently aggregate behaviour profiles across devices. Compared with traditional machine learning, in the works described using deep learning, features are often automatically extracted from raw device traffic.

(3) *Device Integrity Testing Based on DL*. Hardware Trojans are a major security concern where hardware can be accessed by untrusted third-party. Based on the availability of trusted (i.e., golden) chips, hardware Trojan detection methods can be split into methods that utilise golden chips and alternative approaches. Traditional methods include one-class anomaly detection, two-class classification, clustering, and outlier-based, utilising training data such as on-chip sensor data and on-chip traffic data.

Research on the topic of deep learning-based hardware Trojan detection methods is limited but increasing, with many currently based on simple neural networks as an anomaly detector. In works such as [21], they use power consumption data as the model input. To reduce the noise in data acquisition, wavelet transforms are used. A neural network is used to distinguish between normal chip power consumption and deviation in chip performance where a Trojan may be present. Wen et al. [22] use self-organizing maps (SOMs) to detect hardware Trojans. They employ Hotspot to catch the steady-state heat-map from running IC. Then, a 2-dimensional principal component analysis (PCA) is used to extract features from the heat-map. The SOM is used to automatically distinguish Trojan-infected chips. Both of these methods can efficiently detect hardware Trojan. Reshma et al. [23] argue that there exists a large intercluster distance between normal nodes and Trojan infected nodes, especially in the controllability and transition probability. They extract features from chips using autoencoders and use k-means to find Trojan nodes. Work by [24] proposes to extract features from netlists; for each netlist, they get 11 features. Then, the deep multilayer neural network is used to find out malicious netlist. However, the role they play is as an anomaly detector with predefined features. It is suggested that further research of deep learning in this application is still required.

3.2. Network Behaviours in IoT. Here, we focus on the modelling of network behaviours as a result of IoT devices, including device access control, connection-related activities, firmware upgrades, and remote access and control of devices. In particular, it would be beneficial to develop a model that can identify malicious behaviours across the network so as to block remote access. The following network activities will be considered: network nefarious activity/abuse, eavesdropping interception/hijacking, outage, damage/loss, and failures/malfunctions. Since IoT devices are typically constrained in terms of computational resource, detectors that are designed to operate on the devices will therefore need to be lightweight and maintain efficiency. Botnet and DDoS are two primary threats that have been observed on IoT networks in recent times, such as the Mirai botnet that managed to access and control millions of low-level devices. As the number of connected IoT devices increases, so will the nature of attacks that attempt to leverage these to conduct large-scale DDoS operations. Deep learning has recently been used to attempt to identify such attacks. Meidan et al. [25] use deep autoencoders to build normal behaviour profiles for each device. They extract statistical traffic features and train autoencoders with features from

benign traffic. When applied to new traffic observations for a new IoT device, there exists a bigger reconstruction error on the trained autoencoder which can be used to indicate that the device could be compromised. Similar approaches used in Kitsune [26] use ensembles of autoencoders to identify anomalies in IoT such as Mirai. Both of the above approaches assume that normal traffic activity can be approximately reconstructed, while an anomaly would cause large reconstruction error. While many detection methods borrow ideas from traditional intrusion detection or anomaly detection methods, the above two methods considered the heterogeneous and resource constraints in IoT environment.

Other methods use the CNN to automatically identify malicious traffic in IoT. In [27], they turn the payload in the traffic packet into a hexadecimal format and visualize it into a 2D image. Then, they employ a lightweight CNN framework called MobileNet to extract features from traffic images and malware classification. To deal with the volume of traffic needed to analyse this in a DDoS setting, in [28], they propose a deep learning lightweight DDoS detection system called LUCID. They exploit the weight sharing properties of the CNN to classify the traffic, which makes it efficient to be deployed in resource-constrained hardware. To efficiently extract features from network traffic, authors in [29] employ damped incremental statistics as basic features. They then use triangle area maps (TAMs)-based multivariate correlation analysis (MCA) to generate grayscale images as training data from normalized traffic features. They then use these as input to a CNN to learn a model for detecting anomalies.

3.3. Model Data Abuse in IoT Environment. Data gathered by IoT networks can be of great value, and abuse of this data can result in serious consequence, e.g., the case made against Cambridge Analytica. It is therefore crucial that IoT devices manage data responsibly. Data leakage can occur from the generation of data, the use of data, and the transmission/storage of data over the IoT network. For example, data collection by smart meters will reflect home usage patterns for electricity, gas, or water, which if leaked could expose attackers to information about when the house is occupied or not. Similarly, this information could be exposed by other smart devices such as kitchen and entertainment appliances. Intelligent IoT services will naturally aim to gather personal information to further inform the services being provided, where personalisation is deemed as enriching user experience. Five context parameters related to IoT data privacy are proposed by [30]: place ("where"), type of collected information ("what"), agent ("who"), purpose ("reason"), and frequency ("persistence"). In this section, we briefly review works related to data privacy and data integrity.

(1) *Data Privacy in IoT with Deep Learning.* In [31], they study visual privacy within an IoT setting. With low-end IoT cameras, they propose a method for constructing privacy protected and forgery-proof high frame-rate videos. They deployed their software prototype on three different IoT settings: on-site, vehicular, and aerial surveillance. In [32], the authors propose a deep and private-feature learning framework called deep private-feature extractor (DPFE). Based on information theoretic constraints, they are training a deep model which allows the user to prevent sharing sensitive information with a service provider and at the same time enables the service provider to extract approved information using the trained model. Similar work in [33] proposes a feature learning framework that leverages a double projection deep computation model (DPDCM). Different from the traditional deep learning framework, they use double projection layers to replace the hidden layers, which can learn interactive features from big data. Furthermore, they design a training algorithm to fit the DPDCM model. To improve the learning efficiency, cloud computation is used. They also propose privacy-preserving DPDCM based on BGV encryption to protect personal data.

(2) *Federated Learning.* Recently, there has been much interest in developing methods where a collective of devices can contribute towards a global shared model, whilst maintaining the privacy of data that is stored locally on each device. This is well suited in settings where there is a large population of devices that would benefit from collective knowledge but where there are not the rights or the ownership of the devices to control the data. Smart phone devices benefit from federated learning for the purpose of improving predictive services (e.g., predictive text and location recommendations) whilst not disclosing information about other mobile phone users. Smart meters and other IoT devices would benefit in a similar manner. Works by [34] demonstrate that decentralized federated learning can improve data privacy and security, while reducing economic cost. Works in [35] integrate deep reinforcement learning algorithms and the federated learning framework into an IoT edge computing system. The main focus of their work is to improve the efficiency of the mobile edge computing system. They design a framework called "In-Edge AI" to maximise the collaborative efficiency among devices and edge nodes. With this framework, learning parameters can be exchanged efficiently for better training and inference. Their framework can reduce unnecessary system communication while at the same time carry out dynamic system level optimization and application-level enhancement. Wang et al. studied a broad range of machine learning models optimized with gradient descent algorithms [36]. Their research first analyses the convergence bound of distributed gradient descent

algorithms. Then, they propose an algorithm to reach the best trade-off between local and global parameter learning while given limited resource budget.

(3) *Data Integrity in IoT with Deep Learning*. In an IoT setting, upholding integrity is vital to ensure that there is consistency between the actual, physical observation, and the transmitted data or signals that represent this activity. False data injection (FDI) is an attack against a cyber-physical system by modification of the sensor data, which could include SCADA (supervisory control and data acquisition) systems used widely in sectors supporting critical national infrastructure. For example, an FDI attack on an engine sensor could cause erroneous sensor outputs which would result in severe impact on physical maintenance algorithms. Likewise, the infamous Stuxnet attack [5] involved FDI to falsify the behaviour of the centrifuges that then caused physical destruction to the premises.

Recently, deep learning has been used in false data injection detection both in Internet and IoT. Works in [37] use deep learning algorithms to learn the behaviour feature model from historical sensor data and employ the learned model to infer the FDI behaviour in real time. Similar work was proposed by Wang et al. WangH2018 used a two-stage sparse scenario-based attack model to detect attack in smart grid given incomplete network information. To effectively detect established cyber-attacks, they develop a defense mechanism based on interval state model. In their model, they use a dual optimization method to model the lower and upper bounds of each state parameter, which will maximise variation intervals of the system variable. At last, they employ the deep learning model to properly learn nonlinear and nonstationary behaviour features from historical electric usage data.

3.4. Deep Learning Methods for IoT Security. In this section, we will summarize the methods using deep learning techniques to enhance IoT security. Building on this, we propose methodologies that can extend towards improved IoT security.

3.4.1. Feature Learning Process. Traditionally, feature extraction consists of data collection, data preprocessing, and feature extraction. For the purpose of our work, we will separate this out further to consider four steps: data collection, data encoding, feature definition, and feature extraction. Figure 2 shows the behaviour analysis using a multilayered hierarchical Bayesian networks, in which security features are categorised into static features, dynamic features, and causal features based on existing features extracted from IoT security behaviour databases.

In the data collection phase, raw data such as RF signals, device features, heat-map, and raw network packets are collected. Raw data can often be very large, of mixed data types, and can contain many unrelated records, and so there is a need to establish how to manage this information. Data encoding is the process of defining the basic element of interest that is contained within the input, such as individual pixels within a given image or individual packets within a network traffic stream. Here, we represent each element as x_i. In the feature definition stage, data are organised such that a coherent understanding of the data object can be analysed. Typically, input elements could be organized as a distribution, a sequence, a matrix, or more recently in deep learning, a tensor. Following data encoding, raw inputs can be transformed into a format that could be used as input for a deep learning model. Here, we define the feature definition process as D, and then the data after feature definition can be represented as

$$X = D(x_1, x_2, \ldots, x_k), \tag{1}$$

where D is the process to organize basic element into predefined orders. Based on feature definition, features are from inputs. Methods such as statistical methods, series analysis, frequency analysis, or machine learning are used to extract features from organized data elements. Here, we define the feature extraction as

$$V = F(X) = F(D(x_1, x_2, \ldots, x_k)), \tag{2}$$

where F is used to represent the feature extraction methods. Usually, the output in feature extraction is feature vectors with fixed length $V = (v_1, v_2, \ldots, v_m)$. In this work, a two-step data preprocessing phase is introduced: (1) a data encoding process that can be used to extract suitable features from mixed raw inputs and (2) a feature definition process that provides structure to our data.

3.4.2. Deep Learning for Device Feature Extraction. IoT networks can consist of a very large number of connected devices, which can mean that identifying a specific device within a network becomes challenging. Here, we focus on techniques to extract features that can identify a specific IoT device.

One of the most powerful aspects of deep learning is the ability to automatically learn useful features from raw inputs, e.g., autoencoders. For device identification, deep learning methods for device feature extraction can be classified based on the raw data that they use. Information such as sensor noise pattern, radio frequency features, or energy consumption can reflect the uniqueness of devices. Using deep learning, higher level features can be extracted and even very subtle differences between devices can be discovered. Such is the case in camera identification, where raw images captured by using the camera can be gathered. Here, we first define the raw image set as I and then extract the noise pattern from images, which are considered to be unique for a given device. Usually, noise patterns can be calculated as follows:

$$N = I - F(I), \tag{3}$$

where I is the original image containing the original noise and $F(i)$ is the denoised version of I. The residual noise N is called signal noise which is typically unique for a given

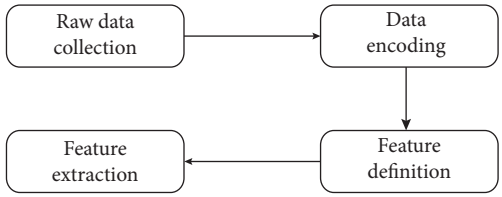

Figure 2: Feature learning processes.

device. To extract signal noise patterns from an image, statistical methods treat the residual noise as a two-dimensional distribution and extract features such as mean, max, skewness, and kurtosis. Using a frequency representation, noise signals can be treated as a two-dimensional signal, and then methods such as wavelet transform or Fourier transform can be used to identify the frequency of noise level. Different from methods above, deep learning methods such as in [10] feed the signal noise matrix directly into the CNN that aims to automatically extract features from noise with minimal human intervention.

Using deep learning, in [11], they learn the signal noise pattern with the signal noise extraction step. For each colour image I, with its camera model L, the authors extract K nonoverlapping patches P_k, $k \in [1, K]$, with each of size 64 ×,64 pixels. To avoid selecting uninformative regions of the image (e.g., dark or saturated pixels), they exclude all regions where the average pixel value is near to half of the image dynamic range. They use the CNN to extract noise feature representation from regions. After that a set of N $(N-1)/2$ linear binary SVMs are trained to identify the difference between different camera models.

Similar work has been performed for RF fingerprinting [13, 38]. In [38], RF signals (IQ) are collected from multiple devices. They consider the Zigbee device baseband as a complex time series represented as follows:

$$r(t) = s_1(t) + n_1(t), \tag{4}$$

where $n(t)$ represents the noise. The training data used here are historical in-phase and quadrature (I and Q) data from six ZigBee devices transmitting at 0, −1, −5, −10, and −15 dBm. They experiment using different window sizes of 16, 32, 64, 128, and 256 which represent the number of I and Q input sequences into the deep learning models. Finally, they utilise different deep learning architectures to assess how they perform for classifying Zigbee devices.

From the view of energy consumption, a heat-map of devices can formulate a normal device template. In this manner, malicious modification of hardware could be detected. In work [22], authors split chips into several equal sized grids. Then, they use a randomly generated "*excitation vector*" to feed the running chip. Finally, for each grid, they measure the steady-state temperature. A 2D PCA is used to identify the feature map from the original heat-map.

To train a device recognition model, data must first be collected from devices. Then, data must be transformed to provide features that can be used as input to a deep learning model. Typical input to the deep learning framework can be matrix-based, sequence-based or statistical-based. Then, with the help of deep learning, a normal template of devices can be formulated.

Traditional ML methods rely on human efforts to extract features which may not easily scale when considering IoT devices [39]. Also, manually curated features may be susceptible to attack or could be manipulated by an attack. Using deep learning techniques such as autoencoders, representative features could be identified automatically, which can be used for fingerprinting devices.

3.4.3. Network Behaviour Modelling with Deep Learning. The basic elements that are often considered for network behaviour modelling are packets, flows, and conversations between communication entities. Unlike other data, data in network traffic are heterogeneous. Basic input in network traffic can be divided into three parts: timestamp, connection identifier, and data description. A packet could therefore be represented as $p = <time, header, content>$. Network behaviour can be formally defined as sequence of packets running between communication nodes:

$$X = (p_1, p_2, \ldots, p_m), \tag{5}$$

where packets are sorted based on timestamps.

Given the heterogeneous nature of a network capture, it can be challenging to extract features directly from a packet sequence. Typically, statistical features are calculated over some short time interval to inform feature representation. Features such as interarrival time, packet length, packet count, bytes send, and bytes received can be extracted. This information can reflect network behaviour property such as communication frequency, traffic volume, and connectivity. Furthermore, these features can reflect buffer size and computational ability and also reflect the services that a device provides. This process can be defined as follows:

$$S = S(X) = S(w_1, w_2, \ldots, w_k), \tag{6}$$

where w_i can be represented as the sequence of packets that fall in the ith time window. Typically, researchers may extract uncorrelated statistical features from time, connection, and content.

Deep learning in network behaviour modelling plays two roles: (1) automatic extraction of high-level features from network traffic and (2) automatic identification of corresponding features across feature dimensions. Behaviour modelling based on deep learning can be defined as

$$V = H(S) = H(w_1, w_2, \ldots, w_k), \tag{7}$$

where H represents the black box, nonlinear function used in deep learning. After that a fixed length behaviour vector to represent network behaviour can be achieved.

As mentioned, features such as interarrival time and packet length could inform of device properties such as buffer size, computational ability, and the services that the device provides. With CNN, LSTM, and other deep learning methods, complex service pattern can be extracted. Such as work in [18], they use interarrival time (IAT) as features to

generate a graph of IAT for 100 packets. The graphs are then treated as images, and all images are converted to a size of 150×150, where they are then given as input to a neural network to identify device behaviour pattern. Research from [19] considers network traffic from devices as sequences of packets. They first split traffic into subflows with fixed time interval T. For each subflow, features related to the number of packets, packet length statistics, and protocol-related features are extracted. Then, a LSTM-CNN cascade model is used to extract high-level features from the whole flow. Both [25, 26] use autoencoders to get normal profiles of IoT devices. Both of them extract packet size, packet count, packet jitter, and packet size from the flow of packets, and then the autoencoder is used to reconstruct the original input to find devices that exhibit deviation in their behaviour.

3.4.4. Proposed Approach

(1) *Semantically Meaningful Device Modelling.* Although deep learning methods may achieve greater accuracy in device identification, semantically meaningful device modelling is still lacking. Since the range of possible IoT devices increases every day, signature-based methods would fail to recognise new device types and provide accurate device property records. Here, we propose a semantically meaningful device identification framework. The main concept is to decouple the feature learning process and devices identification process via a middle-layer process called service identification or functional identification. For example, a functional collection may consist of the following activities: {*capture image, record video, share photo, make phone call, play music*}. We could then represent a camera and a smart phone as follows:

(1) Camera (1, 1, 1, 0, 0)
(2) Smart phone (1, 1, 1, 1, 1)

From the example above, it can be inferred that semantic representation would be more robust for device modelling. Even though devices may be updated frequently, the basic functionality that such devices offer will change at a much slower rate. Using deep learning, a three-step device identification framework can be introduced: (1) using deep learning, basic features of devices can be extracted; (2) from the features extracted, the functional or behaviour model can be constructed, and the output of this step would be the functions or services that the device provides; and (3) using the inferred service type to identify the device. The training process of our framework can be described formally as follows:

Step 1. Extract data from devices including signal noise pattern, network traffic, device heat-map, and other relevant attributes. This data are then transformed to a tensor input for a deep learning model. We define the raw data of the ith device as d_i, the training data sets as $D = d_i, d_2, \cdots, d_n$, the device type of ith device as y_i, the training labels of devices as $Y = y_1, y_2, \cdots, y_n$, and the function set of devices as $A = a_1, a_2, \cdots, a_k$ where k is the size of function set.

Step 2. Use deep learning to extract features from raw data, where the feature extraction process can be defined as Φ, and the extract features are defined as $f_i = \Phi(d_i)$, where f_i is the vector of dimension m. The feature extraction process can be seen as a nonlinear mapping from n dimensional raw data space to m dimensional feature space.

Step 3. Find the mapping between features and attributes. Here, the mapping can be a linear or nonlinear decision or even based on a decision tree. The mapping here is defined as Ψ, which maps input of m dimension into a k dimension binary vector, e.g., (1, 0, 1, 0, 0).

Step 4. Based on attributes, use Bayesian theorem to calculate the most possible device type.

Compared against previous deep learning approach, our method clearly separates feature learning and device type recognition using an intermediate layer. Whilst features of a device type may change rapidly, the functions served may change much slower, and so by using a functional (or service) layer, we can develop a much more robust framework for device identification.

(2) *Behaviour Analysis via Multilayered Hierarchical Bayesian Network.* Compared with traditional approaches, deep learning can learn to extract features from a collection of basic statistics automatically. One advantage is that deep learning can find high-level complex features which may be hard to identify in a statistical setting. Similarly, as feature extraction can be used in device identification, network behaviour modelling using deep learning can be considered using the same approach, with the main difference being the product of the deep learning model. In IoT network behaviour analysis, the aim of deep learning is to identify patterns in network behaviour. Using this network behaviour model, malicious behaviours can then be identified at the network level.

However, it is important to note that deep learning models may fail to capture the causal relationship between traffic features and traffic behaviour. Consequently, the inner relationships between features and behaviour may be overlooked by the training process, which is clearly a vital aspect when examining the network characteristics. Here, we follow the approach in our previous work [40], as illustrated in Figure 3, where

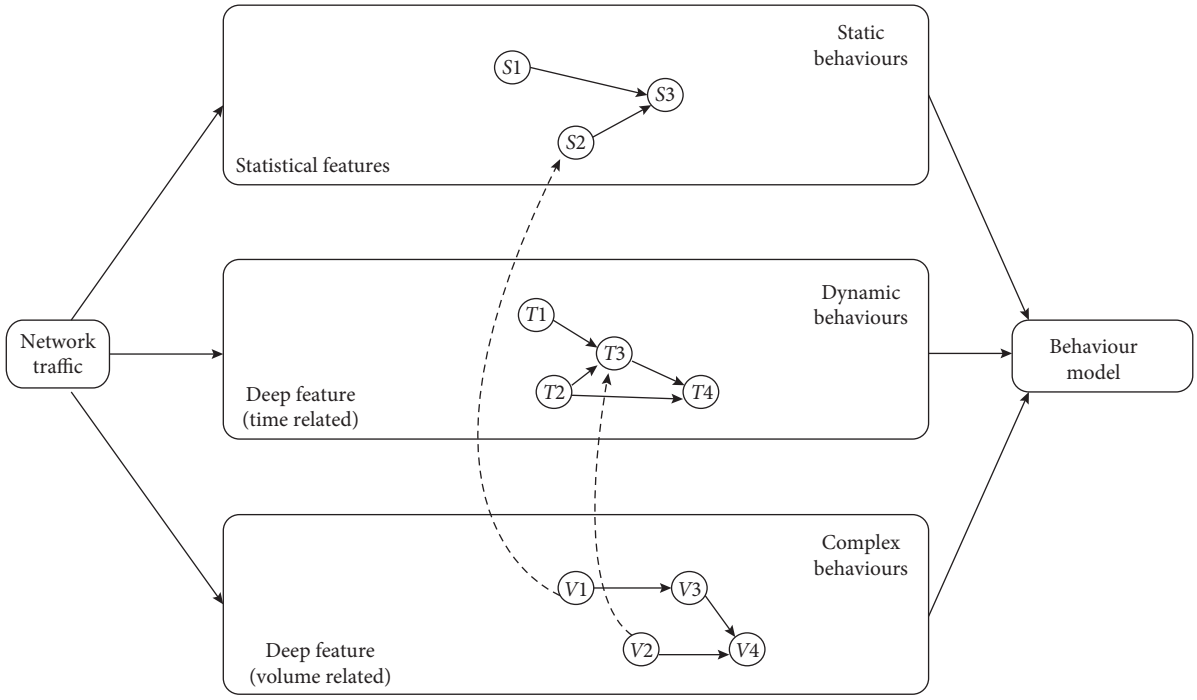

Figure 3: Behaviour analysis via multilayered hierarchical bayesian networks.

features may be extracted using three approaches simultaneously: statistical, time-related, and volume-related.

Then, using a three-layer Bayesian network, we can capture interactions between these different feature representations.

4. Evaluation

To evaluate the performance of DL-based methods, we need a set of performance measures and we need some benchmark dataset to verify the deep learning framework.

4.1. Evaluation Measure. Precision is one of the most common used measures in machine learning. Precision is defined as the number of true positive predictions over the total number of positive predictions made (where either true or false). In an intrusion detection system, f_p would be the number of false alerts.

$$\text{Precision} = \frac{t_p}{t_p + f_p}, \tag{8}$$

where t_p represents the number of correct cases that were classified as positive examples and f_p represents the number of incorrect cases that were classified as positive. Another common performance measure is recall (also known as sensitivity). Recall is defined as the number of true positive predictions over all positive instances. Recall can be defined as follows:

$$\text{recall} = \frac{t_p}{t_p + f_n}, \tag{9}$$

where f_n is the number of positive instances which are incorrectly classified as negative (false negatives). In an intrusion detection system, f_n would be the number of attacks that go undetected. Often, there is an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other.

In binary classification, usually the F1 score (also known as F-Measure or F-score) is used to measure accuracy. It combines the precision and recall at the same time to compute the final score. The F1 score is defined as the harmonic mean of precision and recall. F1 score is defined as follows:

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{10}$$

The maximum score for the F1 measure is 1, where essentially both perfect precision and perfect recall are both achieved.

The receiver operating characteristic (ROC) curve is a graphical plot that can help us find the discrimination threshold of a binary classification system. It is a graphical plot created by plotting the true positive rate against the corresponding false positive rate with different threshold setting. The area under curve (AUC) is the performance measure used for classification systems with different threshold settings. The AUC represents the size of area under the ROC curve, which can be seen as a measurement of separability. It can give us information about how much the learning model is capable of distinguishing between classes. Higher the AUC, the better the classifier can predict a result that matches with the

provided labels. Essentially, the higher the AUC, the higher the model's classification power.

4.2. Evaluation Dataset

(1) *Device Identification.* Miettinen et al. [17] introduce the *captures _IoT Sentinel* dataset. They collect network traffic during the setup stage of 31 smart home IoT devices across 27 different device types (4 types are represented by 2 devices each). They repeat the collection process 20 times for each device type.

The IEEE Signal Processing Society [39] provides a camera model identification dataset, where images exhibit noise patterns that are specific to camera models. They use 10 different camera models to produce images for the training set. For each camera model (one device per camera model), they take 275 full images. They use the same camera models to create a test set but using different devices. Each instance of the test data consists of a 512×512 pixel image that is cropped from the centre of an image taken with the specific camera device.

(2) *Intrusion Detection and Anomaly Detection.* The Bot-IoT dataset [41] was created by researchers at UNSW Canberra Cyber. They simulated attacks including DDoS, DoS, OS and service scans, keylogging, and data exfiltration attacks in a designated realistic network environment. The dataset incorporates a combination of both normal and malicious traffic. The dataset is provided in various data format, such as original pcap files, the generated Argus files, and extracted features in csv format. To assist with the labelling process, they separate the data based on attack category and subcategories.

IoT-23 [42] is a dataset created by researchers at the Stratosphere Laboratory to model realistic network traffic behaviour from IoT devices. The dataset contains 23 captures, with 3 captures representing benign IoT traffic and 20 representing malware traffic captures. For each malicious packet capture, they execute a specific malware sample on a Raspberry Pi, with each malware using several protocols to perform various actions. Samples in their dataset include raw pcap files and Bro generated json files. They separate the dataset into different folders to provide labels.

AWID [43] is a dataset created for wireless network intrusion detection. It contains traffic collected from real-world WiFi traffic. Kolias et al. [43] apply several traditional supervised machine learning approaches to perform intrusion detection on the AWID dataset. They use mutual information to select the top 20 most informative features to then train 8 classifiers. Result shows that their framework can achieve an overall accuracy ranging from 89.43% to 96.2%. In the original AWID dataset, each data instance includes 155 features along with the corresponding training label.

(3) *Data Privacy.* Lee et al. investigated 14 IoT scenarios in [30] to randomly structure five contextual parameters to create IoT scenario descriptions. They request for users to describe their opinions using the free text field. They then use cluster analysis to infer how these five contextual parameters affect people's reaction in IoT environments.

An aggregated electronic signals dataset, namely, REDD [44], was established to infer devices behind the smart meter, which is also known as the task of energy disaggregation. The REDD dataset consists of whole-home and circuit/device-specific electricity consumption for a number of real houses over a period of several months. With this data, the researchers were able to identify the correspondence between devices and electricity usage.

5. Challenges and Research Trends

5.1. Research Challenges

(1) *Efficiency.* Resource constraints of IoT devices remain an important impediment towards deploying deep learning models. Memory efficiency and time efficiency would be two core concerns in implementing deep learning in real IoT systems. Although deep learning models can be trained offline, how to deploy the model remains to be a problem.

The power of a deep learning model originates from the large amount of nonlinear, stacked neurons used in the deep learning architecture. Deep learning models consume raw data which pass through layered neurons to inform a decision. How to reduce the storage and computation needed for execution of the deep learning model in resource constraints applications is an ongoing challenge.

With the development of deep learning methods, various new architectures surpass state-of-the-art performance. However, many of them have not necessarily been developed for the IoT setting. To fully adapt these algorithms to an IoT setting would certainly help to improve the performance of recent studies [45, 46].

(2) *Adaptive.* Devices and applications in the IoT ecosystem are evolving every day, and so deep learning must also be adaptable in the same way. In a real-world network, zero-day attacks will occur. New devices are introduced into the IoT system subsequently. Also, the distribution of network traffic or signal frequency would likely change as new devices join the network. A static trained model cannot easily adapt to changing conditions and so could result in an increase in false positives and false negatives. Another ever-changing element is the request from the end user. Those changes bring new challenges to deep learning applications in the IoT setting. Deep learning algorithms must cope with the fast-evolving environment both from the macro- and micro-perspective.

Another consideration is that many IoT devices may be deployed in a wide scale of areas. The properties of the environment where IoT are deployed may vary from each other. Retraining a deep learning model for each setting not only costs a lot of time but also requires further labelled training data.

(3) *Heterogeneous Data.* IoT devices produce a lot of data with different type and scale, such as data from signal frequency and network traffic, which although they may originate from the same device, they will have different formats. Even data of same type may differ in scale, such as packets number and bytes number. Although they all belong to network features, they use different scale. How to handle those heterogeneous data is an ongoing problem [47, 48].

5.2. Research Trends

(1) *Resource Efficient Deep Learning.* Here are two ways toward resource efficient deep learning: (1) modification on deep learning model itself, compressing or pruning the original deep learning model and (2) result cache, preventing duplicate computation by sharing result among devices.

Previous illuminating studies on neural network focused on compressing dense parameters matrices into sparse matrices. One possible approach to reduce model complexity would be to convert parameters into a set of small dense matrices. A small dense matrix does not require additional storage for element indices and is efficiently optimized for processing.

The ultimate goal of the deep learning model is to inform decisions. One question is if it is needed to make decision for every event in system. One observation shows that device with more computation power would convey richer services, while computation limited devices would be inclined to do a limited set of jobs. So instead, would it be possible to cache the result instead of repeatedly calculating the same decision? Similar ideas have been widely used in computer architecture and operation system design. Methods such as latest recently used (LRU) have long been used in the operating system to avoid duplicate storage access, which could reduce large amount of unnecessary computation.

(2) *Lifelong Learning.* Human and animals have the ability to quickly adapt to new environments; they can continually acquire, fine-tune, and transfer knowledge and skills throughout their lifespan. This ability, known as the ability of lifelong learning, is mediated by a rich set of neuron cognitive mechanisms that together contribute to the development and specialization of our sensorimotor skills as well as to long-term memory consolidation and retrieval. Consequently, lifelong learning capabilities are crucial for computational learning systems and autonomous agents interacting in the real world and processing continuous streams of information. In the IoT setting, with dynamically changing environments and low-powered devices, lifelong learning is needed to create more intelligent and efficient agents.

However, lifelong learning remains a long-standing challenge in machine learning. The most common phenomenon in lifelong learning using traditional machine learning algorithms is called catastrophic forgetting, which means with continual acquisition of incrementally available data from unknown nonstationary data distributions will decrease the performance of learning algorithms. This breaks the basic assumption of deep learning or other machine learning, which needs a stationary data distribution in training data. To improve scaling of deep learning algorithms in IoT setting, lifelong learning is needed to co-operate with information incrementally available over time.

6. Conclusion

In this survey, it is seen that deep learning offers significant potential across the IoT setting. This survey focuses primarily on the use of deep learning technology to investigate the security features of devices in the context of IoT. Specifically, deep learning-based device profiling and fingerprinting were comprehensively discussed. An approach for semantically meaningful device modelling was proposed using a functional layer to improve feature mapping for device identification. Finally, we discussed challenges and research trends that we intend to explore in our future research.

Data Availability

No data can be shared publicly.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G. Egham, "5.8 billion enterprise and automotive IoT: endpoints will be in use in 2020," 2020, https://www.gartner.com/en/newsroom/pressreleases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotiveio.

[2] F.ortune Business Insights, "COVID-19 impact: high dependency on novel technology to bode well for market," 2020, https://www.fortunebusinessinsights.com/industry-reports/internetof-things-iot-market-100307.

[3] Wikipedia, "2016 dyn cyberattack," 2020, https://en.wikipedia.org/w/index.php?title=2016%20Dyn%20cyberattack&oldid=763071700.

[4] A. Greenberg, "Hackers remotely kill a jeep on the highway—with me in it," 2020, https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

[5] R. Langner, "Stuxnet: dissecting a cyberwarfare weapon," *IEEE Security & Privacy Magazine*, vol. 9, no. 3, pp. 49–51, 2011.

[6] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the internet of things (IoT)," *IEEE Internet Initiative*, vol. 1, no. 1, pp. 1–86, 2015.

[7] S. Zhao, S. Li, L. Qi, and L. D. Xu, "Computational intelligence enabled cybersecurity for the internet of things," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 666–674, 2020.

[8] S. Li, T. Qin, and G. Min, "Blockchain-based digital forensics investigation framework in the internet of things and social systems," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1433–1441, 2019.

[9] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[10] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," in *Proceedings of the 2016 IEEE International workshop on information forensics and security (WIFS)*, pp. 1–6, IEEE, Abu Dhabi, UAE, 2016.

[11] L. Bondi, L. Baroffio, and D. Guera, "Etc. first steps toward camera model identification with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 259–263, 2016.

[12] S. Qi, Z. Huang, and Y. Li, "Etc. audio recording device identification based on deep learning," in *Proceedings of the 2016 IEEE International Conference on Signal and Image Processing (ICSIP)*, pp. 426–431, IEEE, Singapore, 2016.

[13] J. Yu, A. Hu, and F. Zhou, "Etc. Radio frequency fingerprint identification based on denoising autoencoders," in *Proceedings of the 2019 international conference on wireless and mobile computing, networking and communications (WiMob)*, pp. 1–6, IEEE, Barcelona, Spain, 2019.

[14] J. Bassey, D. Adesina, and X. Li, "Etc. Intrusion detection for IoT devices based on RF fingerprinting using deep learning," in *Proceedings of the 2019 fourth international conference on fog and mobile edge computing (FMEC)*, pp. 98–104, IEEE, Rome, Italy, 2019.

[15] Y. Meidan, M. Bohadana, and A. Shabtai, "Etc. ProfilioT: a machine learning approach for iot device identification based on network traffic analysis," in *Proceedings of the Proceedings of the Symposium on Applied Computing*, Article ID 506509, Pisa, Italy, 2017.

[16] B. Bezawada, M. Bachani, and J. Peterson, "etc. iotsense: behaviour al fingerprinting of iot devices," 2018, http://arxiv.org/abs/1804.03852.

[17] M. Miettinen, S. Marchal, and I. Hafeez, "Etc. Iot sentinel: automated devicetype identification for security enforcement in iot," in *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 2177–2184, IEEE, Atlanta, GA, USA, 2017.

[18] S. Aneja, N. Aneja, and M. S. Islam, "IoT device fingerprint using deep learning," in *Proceedings of the 2018 IEEE international conference on internet of things and intelligence system (IOTAIS)*, pp. 174–179, IEEE, Bali, Indonesia, 2018.

[19] L. Bai, L. Yao, and S. S. Kanhere, "etc. Automatic device classification from network traffic streams of internet of things," in *Proceedings of the 2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pp. 1–9, IEEE, Chicago, IL, USA, 2018.

[20] H. Gao, C. Liu, Y. Li, and X. Yang, "V2VR: reliable hybrid-network-oriented V2V data transmission and routing considering RSUs and connectivity probability," *IEEE Transactions on Intelligent Transportation Systems*, p. 1, 2020.

[21] X. Li, F. Xiao, and L. Li, "Etc. detection method of hardware trojan based on wavelet noise reduction and neural network," in *Proceedings of the International Conference on Cloud Computing and Security*, pp. 256–265, Springer, Cham, Switzerland, 2018.

[22] H. Gao, L. Kuang, Y. Yin, B. Guo, and K. Gou, "Mining Consuming behaviour s with temporal evolution for personalized recommendation in mobile marketing apps," *ACM/Springer Mobile Networks and Applications (MONET)*, vol. 25, no. 4, pp. 1233–1248, 2020.

[23] K. Reshma, M. Priyatharishini, and M. Nirmala Devi, "Hardware trojan detection using deep learning technique," *Advances in Intelligent Systems and Computing*, Springer, Singapore, pp. 671–680, 2019.

[24] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Hardware trojans classification for gate-level netlists using multi-layer neural networks," in *Proceedings of the 2017 IEEE 23rd international symposium on on-line testing and robust system design (IOLTS)*, pp. 227–232, IEEE, Thessaloniki, Greece, 2017.

[25] Y. Meidan, M. Bohadana, Y. Mathov et al., "N-BaIoT-Network-Based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[26] H. Gao, W. Huang, and Y. Duan, "The cloud-edge based dynamic reconfiguration to service workflow for mobile ecommerce environments: a QoS prediction perspective," *ACM Transactions on Internet Technology*, 2020.

[27] R. Shire, S. Shiaeles, and K. Bendiab, "Etc. malware squid: a novel iot malware traffic analysis framework using convolutional neural network and binary visualisation," *Smart Spaces, and Next Generation Networks and Systems*, p. 6576, Springer, Cham, Switzerland, 2019.

[28] R. Doriguzzi-Corin, S. Millar, and S. Scott-Hayward, "Etc. LUCID: a practical, lightweight deep learning solution for DDoS attack detection," *IEEE Transactions on Network and Service Management*, vol. 17, 2020.

[29] J. Liu, S. Liu, and S. Zhang, "Detection of IoT botnet based on deep learning," in *Proceedings of the 2019 Chinese control conference (CCC)*, pp. 8381–8385, IEEE, Guangzhou, China, 2019.

[30] H. Lee and A. Kobsa, "Understanding user privacy in internet of things environments," in *Proceedings of the 2016 IEEE 3rd world forum on internet of things (WF-IoT)*, pp. 407–412, IEEE, Reston, VA, USA., 2016.

[31] H. Yu, J. Lim, K. Kim et al., "Pinto: enabling video privacy for commodity IoT cameras," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1089–1101, Toronto Canada, 2018.

[32] S. A. Osia, A. Taheri, A. S. Shamsabadi et al., "Deep private-feature extraction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 1, pp. 54–66, 2018.

[33] Q. Zhang, L. T. Yang, Z. Chen et al., "Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2896–2903, 2017.

[34] Y. Zhao, M. Li, L. Lai et al., "Federated learning with non-iid data," 2018, http://arxiv.org/abs/1806.00582.

[35] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.

[36] S. Wang, T. Tuor, T. Salonidis et al., "Adaptive federated learning in resource constrained edge computing systems,"

IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1205–1221, 2019.

[37] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: a deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, 2017.

[38] H. Jafari, O. Omotere, and D. Adesina, "Etc. IoT devices fingerprinting using deep learning," in *Proceedings of the MILCOM 2018-2018 IEEE military communications conference (MILCOM)*, pp. 1–9, IEEE, Los Angeles, CA, USA, 2018.

[39] IEEE Signal Processing Society, "IEEE's signal processing society camera model identification," 2020, https://www.kaggle.com/c/sp-society-camera-modelidentification.

[40] S. Li, T. Tryfonas, G. Russell, and P. Andriotis, "Risk assessment for mobile systems through a multilayered hierarchical bayesian network," *IEEE Transactions on Cybernetics*, vol. 46, no. 8, pp. 1749–1759, Aug. 2016.

[41] 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.

[42] Stratosphere Laboratory, *A Labeled Dataset with Malicious and Benign IoT Network Traffic*, Stratosphere Laboratory, Prague, Czechia, 2020.

[43] C. Kolias, "Intrusion detection in 802.11 networks empirical evaluation of threats and a public dataset," *IEEE Communications Surveys & Tutorials*, vol. 18, pp. 184–208, 2015.

[44] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Proceedings of the Workshop on data mining applications in sustainability (SIGKDD)*, vol. 25, pp. 59–62, San Diego, CA, USA, 2011.

[45] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: current status, challenges and prospective measures," in *Proceedings of the 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 336–341, IEEE, London, UK, 2015 December.

[46] K. Ashton, "That internet of things thing," *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[47] H. Wang, J. Ruan, G. Wang et al., "Deep learning-based interval state estimation of AC smart grids against sparse cyber attacks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4766–4778, 2018.

[48] A. Ukil, S. Bandyopadhyay, and A. Pal, "Privacy for IoT: involuntary privacy enablement for smart energy systems," in *Proceedings of the 2015 IEEE international conference on communications (ICC)*, pp. 536–541, IEEE, London, UK, 2015.