**Survey on IoT Security Analysis Using Lightweight Machine Learning**

Yaregal Tessema1  
*Department of Software Engineering, College of Electrical and Mechanical Engineering  
Addis Abeba Science and Technology University*Addis Abeba, Ethiopia  
 yayasoles@gmail.com

Tulu Tilahun Hailu1,2  
*1. Artificial Intelligence and Robotics Center of Excellence, Addis Ababa Science and Technology University, Addis Ababa, Ethiopia*  
*2.Department of Software Engineering, College of Electrical and Mechanical Engineering, Addis Ababa Science and Technology University, Addis Ababa,Ethiopia*

*Abstract*— Applications for the Internet of Things (IoT) have been employed in a wide range of industries, including smart homes, healthcare, smart energy, and Industrial 4.0. IoT offers many advantages, such as ease and efficiency, but it also adds a lot of new risks. The problem is frequently made worse by the potential number of linked IoT devices and the ad hoc nature of such systems. The management of IoT now faces substantial issues in the areas of security and privacy. Recent research has shown that deep learning algorithms are quite effective and have numerous advantages over previous approaches for doing security analysis of IoT devices. In order to address security and privacy concerns, this study seeks to present a detailed assessment of deep learning applications in IoT.

Keywords—IoT, Machin Learning, Light weight, data privacy

# Introduction

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, on objects, animals, or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction". The IoT encompasses a large range of devices ('things'), among which everyday household electronics, such as dishwashers, fridges, smart cameras, smart watches, smart glasses, smart TVs, and smart light bulbs.

Security in IoT software is not a joke anymore since the IoT systems are improving our day-to-day life in a way that our day-to-day life depended on it. In this rise of IoT, where billions of devices are expected to be integrated into horizontal applications [4] [5]. Security problems are massively increasing because the amount of linked smart devices constantly grows with their use of different standards, the heterogeneity of the devices, their different implementation way [6]- [8], making the security testing operations daunted. Imagine your smart TV is hacked by someone and its record audio in your Salon or bedroom, imagine your IoT connected car is hacked, or an intruder successfully hacked your IoT supported door, from these simple cases you can imagine the consequence that successful penetration of security holes in an IoT supported devices. Since every device such as your watch, washing machine, your doorbell, your ovens, etc. is being able to connect to the internet which increases your vulnerability to security.

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 [6], 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 [7]. To be efficient and lightweight to deploy, many IoT applications run on embedded CPUs with limited 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 [8] 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.

Given the complexity of developing IoT systems integration, the device limitations 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, I study how machine learning can be used to enhance security in the IoT industry. Firstly, I review security and privacy concerns in IoT systems. I 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, I present the research Gaps that I have identified. The main contributions of this paper are summarized as follows:

1. I summarize and provide a taxonomy of recent work using deep learning to enhance the security property of IoT systems and how deep learning can help to detect security attacks.
2. I identify the weaknesses that still exist in current research and the discrepancies between these weaknesses and the requirements of the IoT setting

# problem statement

Over the last few years, IoT devices and IoT-enabled solutions have become significantly popular both for consumers and industries. IoT is not just about embedded devices, but also comprises an ecosystem of device hardware, system integration, connectivity, data storage, security, IoT platform providers, IT and communication service providers, and application development. Currently, there are more IoT devices connected to networks than the number of human beings on the earth. These IoT devices carry a lot of sensitive data which remains insecure. IoT security has become the subject of strong consideration after several high-profile incidents where a common. IoT device was used to infiltrate and attack a larger network, hacking of internet-connected devices, surveillance concerns, and privacy.

As said in the introduction part IoT security is still a sensitive issue several studies have been made in the past years which focuses on developing a testing strategy, testing the devices for security hole and using deep learning to test the security of the devices. Most the strategies are not successful

Give the IoT devices limitation such as computation limitation, storage limitation, power limitation and the way they need to work such as they have to communicate

# Litrature review

Machine learning (ML) is considered to be the founding pillar of Modern synthetic intelligence [10]. ML has been widely used in computer vision, speech recognition, robotics, and many other application areas and achieved an extraordinary result Compared with other techniques. Machine learning has some key advantages. With billions of devices interconnected together to gather and transfer data worldwide, IoT systems naturally produce a large amount of data. Deep learning has significant potential to help asses’ user’s behaviors in complex IoT systems. Furthermore, deep learning could enable IoT devices to learn complicated behavioral patterns more effectively than traditional learning techniques.

The IoT is a complete ecosystem that contains a heterogeneous devices and connections, a huge number of users, and a large amount 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 behaviors exhibited in the ecosystem so focusing on the individual device or layers security will hard to do and do not grant the security of the overall ecosystem of the IoT. I grouped the IoT security issues in to three categories:

1. To identify the distinctiveness of each IoT device in the network.
2. To investigate the network behaviors in IoT.

## To identify the distinctiveness of each IoT device in the network.

Each and every device in the IoT ecosystem will often have a fixed features such as physical characteristics or services that it provides, power they consume and etc. based on this futures it is possible to profile a device to uniquely identify the device from other devices found in the same IoT ecosystem.

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 analyzed 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: in this regard I classified the devices the papers in to three categories which are (1), Device Identification Using DL, (2) Service Fingerprint Extraction Using DL, (3) Device Integrity Testing Based on DL. Let’s see the papers done in each category one by one

### Device Identification Using DL

Traditional methods of IoT device identification are by using 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 characterize data and therefore could be effective for device identification. Deep learning methods can extract features from the signal or traffic produced by the device in order to recognize and identify the device. Work in [11] 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 the 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 [12] 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 [13]. Radio fingerprinting has also been studied where devices are identified by their wireless radio device properties. In [14], 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 [15]. 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.

### 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 behavior 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, web traffic and or their battery consumptions as inputs to extract behavior al fingerprints.

Previously, researchers have employed machine learning to address challenges in IoT [16–18]. Meidan et al. propose an IoT device classification framework based on HTTP packet analysis [16]. 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 [17], the authors propose to approximately model IoT behavior 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 [18], 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 behavior fingerprinting. Reference [19] 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 [20] attempts to automatically identify the semantic type of a device by analyzing its network traffic. First, they define a collection of discriminating features from raw traffic flows, and those features are used to characterize 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 [21] 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 behavior profile with minimal human efforts. Federated learning is utilized in DI¨OT to efficiently aggregate behavior profiles across devices. Compared with traditional machine learning, in the works described using deep learning, features are often automatically extracted from raw device traffic.

### 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 utilize golden chips and alternative approaches. Traditional methods include one-class anomaly detection, two-class classification, clustering, and outlier-based, utilizing 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 [22], 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. [23] 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. [24] 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 [25] 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.

## Network behaviors in IoT

Here, I focus on the modelling of network behaviors 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 behaviors 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. [26] use deep autoencoders to build normal behavior 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 [27] 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 [28], 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 Mobile Net to extract features from traffic images and malware classification. To deal with the volume of traffic needed to analyses this in a DDoS setting, in [29], 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 [30] 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.

## 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 Analytical. 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 [31]: place (“where”), type of collected information (“what”), agent (“who”), purpose (“reason”), and frequency (“persistence”). In this section, I briefly review works related to data privacy and data integrity

### 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

### 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 maximize 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.

### 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 behavior 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 behavior feature model from historical sensor data and employ the learned model to infer the FDI behavior 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 maximize variation intervals of the system variable. At last, they employ the deep learning model to properly learn nonlinear and nonstationary behavior features from historical electric usage data.

## Deep Learning Methods for IoT Security

In this section, I will summarize the methods using deep learning techniques to enhance IoT security. Building on this, I propose methodology that can extend towards improved IoT security. There are three major techniques of deep learning used for IoT security such as Feature Learning Process, Deep Learning for Device Feature Extract and Network Behavior Modelling with Deep Learning. Let’s see their respective concepts and papers done on each.

### Feature Learning Process

Traditionally, feature extraction consists of data collection, data preprocessing, and feature extraction. Sometimes data processing can split into data encoding, feature definition as shown in figure below

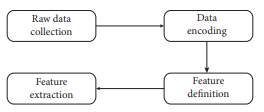


Figure 1 Feature Learning Process

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, data are organized such that a coherent understanding of the data object can be analyzed. 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. Methods such as statistical methods, series analysis, frequency analysis, or machine learning are used to extract features from organized data element

### 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, I 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 e, I 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 following



Equation 1

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 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 [32], they learn the signal noise pattern with the signal noise extraction step. For each color image I, with its camera model L, the authors extract K none overlapping patches Pk, k ∈ [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 [33, 34]. In [34], RF signals (IQ) are collected from multiple devices. They consider the ZigBee device baseband as a complex time series represented as follows:



* 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 utilize 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 [35], 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 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 [36]. 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.

### Network Behavior Modelling with Deep Learning.

The basic elements that are often considered for network behavior 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 behavior can be formally defined as sequence of packets running between communication nodes:



Equation 3

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 behavior 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



Where wi 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 behavior modelling plays two roles: (1) automatic extraction of high-level features from network traffic and (2) automatic identification of corresponding features across feature dimensions. Behavior modelling based on deep learning can be defined



Where H represents the black box, nonlinear function used in deep learning. After that a fixed length behavior vector to represent network behavior 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 [37], they use interarrival time (IAT) as features

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 behavior pattern. Research from [38] considers network traffic from devices as sequences of packets. They first split traffic into sub flows with fixed time interval T. For each sub flow, 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 [39, 40] 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 behavior.

# Research gaps

**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 [41, 42].

**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.

**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 [43, 44].

**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.

**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.

# Conclusion

According to this survey, deep learning has demonstrated significant potential in the IoT environment. The investigation of IoT device security aspects using deep learning technology is the main emphasis of this survey. Deep learning-based device fingerprinting and profiling in particular were thoroughly covered. To enhance feature mapping for device identification, a method for semantically meaningful device modeling was put forth. Finally, I talked about the issues and research directions I want to investigate in our future work.

##### References

1. C.H.R.I.S.T.O.P.H.E.R.S.Y.O.O. (n.d.). article on IOT. Cigionline.Org. Retrieved January 31, 2022, from https://www.cigionline.org/articles/emerging-internet-things/?utm\_source=google\_ads&utm\_medium=grant&gclid=Cj0KCQiArt6PBhCoARIsAMF5wagddds86MyMXi-niFl-yx3swGYbPzhqvyGooatrEBmvZdKB-NKO\_jQaAneLEALw\_wcB
2. B.L.E.G.E.A.R.D. (2019, March 25). Security Testing of IoT systems. In C.V.I.H.O., B.O.U.Q.U.E.T.F.A.B.R.I.C.E., & F.L.E.G.A.L.L. (Eds.), Model-Based Testing for IoT Systems - Methods and Tools (1st ed., Vol. 1, pp. 24–32).
3. Ahmad, A., Baqa, H., Hwang, J., & le Gall, F. (2018, February). IoT-TaaS: Towards a Prospective IoT Testing Framework. 10.1109/ACCESS.2018.2802489, IEEE Access, 1, 1–14.
4. L. Ericsson, “More than 50 billion connected devices,” White Paper, 2011
5. S. Ziegler, C. Crettaz, L. Ladid, S. Krco, B. Pokric, A. F. Skarmeta, A. Jara, W. Kastner, and M. Jung, “IoT6–moving to an ipv6-based future IoT,” in The Future Internet Assembly, pp. 161–172, Springer, 2013.
6. Wikipedia, “2016 dyn cyberattack,” 2020, https://en. wikipedia.org/w/index.php?title=2016%20Dyn%20cybera ttack&oldid=763071700.
7. 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/.
8. R. Langner, “Stuxnet: dissecting a cyberwarfare weapon,” IEEE Security & Privacy Magazine, vol. 9, no. 3, pp. 49–51, 2011.
9. 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.
10. 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.
11. 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.
12. 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.
13. 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
14. 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.
15. 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.
16. 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.
17. B. Bezawada, M. Bachani, and J. Peterson, “etc. iotsense: behavior al fingerprinting of iot devices,” 2018, http://arxiv. org/abs/1804.03852.
18. 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.
19. 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.
20. 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.
21. H. Gao, C. Liu, Y. Li, and X. Yang, “V2VR: reliable hybridnetwork-oriented V2V data transmission and routing considering RSUs and connectivity probability,” IEEE Transactions on Intelligent Transportation Systems, p. 1, 2020.
22. 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.
23. H. Gao, L. Kuang, Y. Yin, B. Guo, and K. Gou, “Mining Consuming behavior 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.
24. 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.
25. 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
26. Y. Meidan, M. Bohadana, Y. Mathov et al., “N-BaIoT-NetworkBased detection of IoT botnet attacks using deep autoencoders,” IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22, 2018.
27. 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.
28. 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.
29. 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.
30. 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.
31. 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.
32. 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.
33. 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
34. 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.
35. 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
36. IEEE Signal Processing Society, “IEEE’s signal processing society camera model identification,” 2020, https://www. kaggle.com/c/sp-society-camera-modelidentification.
37. S. Aneja, N. Aneja, an d 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.
38. 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.
39. Y. Meidan, M. Bohadana, Y. Mathov et al., “N-BaIoT-NetworkBased detection of IoT botnetattacks using deep autoencoders,” IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22,2018.
40. 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.
41. 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.
42. K. Ashton, “that internet of things thing,” RFID Journal , vol. 22, no. 7, pp. 97–114, 2009.
43. 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.
44. A. Ukil, S. Bandyopadhyay, and A. Pal, “Privacy for IoT: involuntary privacy enablement forsmart energy systems,” in Proceedings of the 2015 IEEE international conference oncommunications (ICC), pp. 536–541, IEEE, London, UK, 2015.