Computer Communications 151 (2020) 495–517

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Review

Deep learning and big data technologies for IoT security

Mohamed Ahzam Amanullah [a,*], Riyaz Ahamed Ariyaluran Habeeb [b,c], Fariza Hanum Nasaruddin [c], Abdullah Gani [d,e], Ejaz Ahmed [e], Abdul Salam Mohamed Nainar [f], Nazihah Md Akim [b], Muhammad Imran [g]

[a] Research & Innovation Development, Telekom Research & Development Sdn. Bhd, Cyberjaya, Selangor, Malaysia
[b] Faculty of Science, Technology, Engineering & Mathematics, International University of Malaya-Wales, Malaysia
[c] Department of Information System, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
[d] Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Sabah, Malaysia
[e] Centre for Research in Mobile Cloud Computing, University of Malaya, Malaysia
[f] Greenview Islamic International School, Shah Alam, Selangor, Malaysia
[g] College of Applied Computer Science, King Saud University, Riyadh, Saudi Arabia

ARTICLE INFO

Keywords:
Deep learning
Big data
IoT security

ABSTRACT

Technology has become inevitable in human life, especially the growth of Internet of Things (IoT), which enables communication and interaction with various devices. However, IoT has been proven to be vulnerable to security breaches. Therefore, it is necessary to develop fool proof solutions by creating new technologies or combining existing technologies to address the security issues. Deep learning, a branch of machine learning has shown promising results in previous studies for detection of security breaches. Additionally, IoT devices generate large volumes, variety, and veracity of data. Thus, when big data technologies are incorporated, higher performance and better data handling can be achieved. Hence, we have conducted a comprehensive survey on state-of-the-art deep learning, IoT security, and big data technologies. Further, a comparative analysis and the relationship among deep learning, IoT security, and big data technologies have also been discussed. Further, we have derived a thematic taxonomy from the comparative analysis of technical studies of the three aforementioned domains. Finally, we have identified and discussed the challenges in incorporating deep learning for IoT security using big data technologies and have provided directions to future researchers on the IoT security aspects.

Contents

1. Introduction 496
2. Motivation and use cases 497
   2.1. SirenJack 497
   2.2. Turning Up the Freeze 497
   2.3. Attack on Dyn 497
   2.4. IoT fish tank 497
   2.5. Hacked baby monitor 498
3. Background 499
   3.1. Deep learning 499
   3.2. Big data technologies 499
   3.3. IoT security 499
      3.3.1. Confidentiality 499
      3.3.2. Integrity 499
      3.3.3. Availability 499
      3.3.4. Authentication 499
      3.3.5. Access control 500

* Corresponding author.
E-mail addresses: ahzam.amanullah@tmrnd.com.my (M.A. Amanullah), riyaz@iumw.edu.my (R.A.A. Habeeb), fariza@um.edu.my (F.H. Nasaruddin), abdullahgani@ums.edu.my (A. Gani), ejazahmed@ieee.org (E. Ahmed), salam_nainar@ieee.org (A.S.M. Nainar), nazihah@iumw.edu.my (N.M. Akim), dr.m.imran@ieee.org (M. Imran).

https://doi.org/10.1016/j.comcom.2020.01.016
Received 28 October 2019; Received in revised form 21 December 2019; Accepted 7 January 2020
Available online 10 January 2020

4. Taxonomy 500
4.1. Deep learning 500
4.1.1. Deep learning architectures 500
4.1.2. Frameworks 502
4.1.3. Model evaluation 502
4.2. IoT security 503
4.2.1. IoT security application areas 503
4.2.2. IoT security attacks 504
4.2.3. Datasets 506
4.3. Big data technologies 507
4.3.1. Apache Hadoop 507
4.3.2. Apache Spark 507
4.3.3. Apache Storm 507
5. State of the art deep learning for IoT security using big data technologies 507
5.1. Deep learning and IoT security 507
5.2. Deep learning and big data technologies 509
5.3. Deep learning and big data technologies for IoT security 510
5.4. Cloud infrastructure for deep learning, big data technologies, and IoT security 511
6. Open challenges and future directions 511
6.1. Security threat detection 511
6.2. Training duration 512
6.3. Time complexity 512
6.4. Computing-in-memory 512
6.5. Computational and energy constraint 513
6.6. Security at edge 513
7. Conclusion 513
Declaration of competing interest 514
Acknowledgements 514
References 514

1. Introduction

The swift growth in emerging technologies such as, sensors, smartphones, 5G communication, and virtual reality leads to innovative applications such as, connected industries, smart city, smart energy, connected automobiles, smart agriculture, connected building complexes, connected health care, smart retail outlets, and smart supply chain, which adversely contribute to the accumulation of massive amounts of data. A study conducted by the National Cable & Telecommunications Association (NCTA) predicts that by 2020, approximately 50.1 Billion Internet of Things (IoT) devices will be connected to the Internet. The growth of IoT devices makes the security of these devices debatable [1,2]

According to McAfee (2018), there has been a barrage of cyberattacks and data breaches that has hit almost every industry since 1st of January 2018. Further, many of these attacks were targeted on IoT devices. The increasing use of IoT devices invites the cybercriminals to target them. Additionally, the prospect of interconnectivity among IoT devices makes them vulnerable [3]. Furthermore, VDC Research Group Inc. have also conducted a study to determine the obstacles in developing connected devices. The research has indicated that 60% of the obstacles are related to security requirements in developing connected devices [4]. In addition, based on Kaspersky Lab's collection, the number of malware samples for IoT devices has seen a rapid increase from 3219 samples for the year 2016 to 121588 samples for the year 2018. It is clearly evident that there are huge number of vulnerabilities for the IoT devices [5].

According to [2], many organizations are exposed to greatest challenges in monitoring network based threats, prominently in the following sectors: government, energy, healthcare, banks, and research centres. Moreover, these sectors invest in security monitoring tools in order to protect and secure their infrastructure. As mentioned earlier, generally, the IoT devices generate immense amounts of data that flows through networks. Data that flows through a network is at the possible risk for network attacks. Further, the study has argued that the existing tools and techniques are insufficient to detect innovative attacks triggered by cybercriminals due to the volume, velocity, variety, and veracity of data. Moreover, when huge amounts of data are being handled by the network, the security analytics report on a weekly or monthly basis would not be sufficient enough to detect and mitigate the attacks. Furthermore, the study has asserted that big data technologies would be able to handle the challenges of the volume, velocity, variety and veracity of the data.

Data is generally categorized as big data based on the properties associated with it, commonly known as the V's of big data [6]. Big data technologies are the tools or technologies used to efficiently process these data. Authors of [7], discuss that enterprises collect security related data for regulatory compliance and post hoc forensic analysis. Furthermore, these large enterprises generate approximately 10 to 100 billion events per day. The authors also assert that existing mechanisms lack processing at large scales and big data analytics have been used to analyse and correlate security-related data efficiently and at unprecedented scales.

In this context, this present study proposes to employ deep learning and big data technologies to strengthen the security of IoT devices. Off late, deep learning has gained recognition due to its non-manual feature engineering, unsupervised pre-training, and compression capabilities, these features make the employability of deep learning feasible even in resource constrained networks. Furthermore, deep learning has been widely implemented because of its self-learning capability, potential to yield highly accurate results, and faster processing time. This is vital, as resource constrained system may run into other issues such as out-of-memory access, unsafe programming languages, and so forth [8].

Most of the existing literature separately focuses on deep learning, big data, and IoT security. Some studies have either focused on deep learning [9,10] or big data [11,12] for IoT security. To the best of our knowledge, none of the existing studies have comprehensively reviewed the feasibility of employing both of these technologies in context of IoT security.

Table 1 summarizes most of the existing recent relevant studies and highlights the research gap. From Table 1, it is concludable that many studies have failed to consider the impact of volume, velocity, variety, and veracity of data generated by IoT devices, as against [2]

who have highlighted the impacts in their study. Hence, inclusion of big data technologies becomes mandatory to address the impact of volume, velocity, variety, and veracity of data generated by IoT devices. Additionally, it is clearly evident in Table 1 that not many studies have focused on deep learning and big data technologies for IoT security.

This paper is intended to guide deep learning, big data, and IoT researchers and developers, to whom IoT security would be of primary concern. The contributions of this paper has been summarized below.

i We identified, and highlighted the key issues of IoT security.
ii We have picked five IoT security use cases where deep learning and big data technologies could be of potential solution.
iii We have surveyed the state-of-the-art researches focused on deep learning, big data technologies, and IoT security, to determine the technical applicability and limitations of these three aforementioned domains.
iv We have developed a thematic taxonomy by extracting valuable information from the state-of-the-art.
v We have analysed existing solutions based on the derived taxonomy.
vi We have highlighted the challenges and have proposed guidelines for future researchers to encourage the successful application of deep learning, big data technologies, and IoT security.

However, this study limits its scope only to deep learning and does not discuss on traditional machine learning algorithms with respect to big data technologies and IoT security. Additionally, this survey also does not go into detail about IoT security for each available smart application area, rather discusses in the networking and communications perspective.

This paper is structured as follows:

Section 2 details the motivation and use cases of, deep learning, big data technologies, and IoT security. Section 3 introduces deep learning, big data technologies, and IoT security. Section 4 provides the thematic taxonomy and discusses its components in detail. Section 5 critically analyses the state-of-the-art studies related to deep learning, big data technologies, and IoT security. Section 6 discusses the challenges and proposes future directions. Finally, Section 7 concludes this present study.

2. Motivation and use cases

In this section we have detailed on the motivation for our study and provided some use case scenarios that motivate the survey of deep learning and big data technologies for IoT security.

IoT devices have seen rapid growth in recent years, which is of a great concern in terms of the security risks associated with them. The rapid growth of these devices and the availability of modern hacking technologies have forced the necessity to ensure that IoT devices are not vulnerable to security breaches. However, as of now, IoT devices have been evidently proven to have security vulnerabilities, such as when IoT devices were compromised with the Mirai malware and were used to attack Dyn, a Domain Name System (DNS) provider. Therefore, it is necessary to come up with new technologies or a combination of existing technologies to secure IoT devices from the attackers.

The IoT security requirements such as confidentiality, integrity, availability, authentication, and access control (see Section 3.3) makes IoT devices unique and challenging especially for developers to come up with sophisticated IoT systems that are resistant to IoT based attacks. This study has been motivated by the fact that big data technologies support these security requirements and deep learning algorithms have been proven effective in security attack detection.

Over the years, deep learning has gained wide recognition among researchers and organizations. Due to the capabilities of deep learning it has been applied in a variety of security domains, such as, [20,21], and [22] to identify security breaches. Furthermore, deep learning has proven its success in IoT security, it has been proven by successful implementation in studies [23,24] and [25].

Besides, big data technologies have also been proven to be effective in processing of various types of data. Studies such as, [26,27] and [28] have shown promising results. However, limited studies have been conducted on processing of IoT security data with big data technologies and deep learning algorithms. From our critical analysis, we were able to identify that only two studies have incorporated deep learning and big data technologies for IoT security, which are [29] and [30]. This scenario has motivated us to conduct this study and we believe this study will motivate future researchers in incorporating the three areas discussed. Fig. 1 illustrates the IoT security use cases with their relationship to big data technologies and deep learning characteristics. Additionally, the use cases have been discussed in the following sub sections.

2.1. SirenJack

A vulnerability in emergency broadcast systems produced by Acoustic Technology Inc. (ATI) was identified by Balint Seeber nicknamed SirenJack, a researcher of Bastille Security. The systems allowed command packet broadcast over the air to be captured, modified and replayed. The flaw was discovered when Seeber was auditing emergency alert systems deployed across San Francisco [31,32]. The SirenJack use case is a type of intrusion detection which can be evaded using deep learning and big data technologies as they have shown promising results in detecting intrusions (see Section 4.2.1).

2.2. Turning Up the Freeze

Turning Up the Freeze was a Distributed Denial-of-Service (DDoS) attack conducted on the environmental control systems in two apartment building in eastern Finland. The DDoS attack disabled all environmental control systems in the two apartments completely, which left the people in the apartment cold. In order to rectify the issue, the systems were rebooted. However, the systems got stuck in an endless loop [33]. Environmental control systems that have processing capabilities will be capable of identifying a DDoS attack effortlessly using deep learning and big data technologies. Few fellow researchers were capable of identifying DDoS attacks using deep learning and big data technologies, as discussed in Section 5.

2.3. Attack on Dyn

A major attack was conducted on Dyn, a leading DNS provider on 21st October 2016. The attack was a major DDoS attack that made approximately 85 major websites such as Netflix, Twitter, PayPal, and Sony PlayStation unresponsive for users. This was a series of three attacks, the first wave of attack affected the East coast, the second wave affected California, the Midwest, and the Europe, the third wave was mitigated by Dyn. The attacks are believed to be conducted by large amounts of IoT botnets that were infected by the Mirai malware [34–36]. This major attack could have been mitigated with the use of deep learning and big data technologies. The DNS provider generally stores log data. These log data could have been efficiently processed by big data technologies and analysed using deep learning algorithms, to identify any type of anomalous behaviour. A proven example would be study [26], where the authors were able to analyse anomalous behaviour using big data technologies and machine learning.

2.4. IoT fish tank

In North America, hackers have used Internet-connected fish tank to hack a casino. The fish tank was equipped with sensors to regulate temperatures, food monitoring, and cleanliness of the tank. Hackers used the fish tank to get into the network. It was reported that 10

Table 1
Summary of recent literature relevant to deep learning, big data technologies, and IoT security.

Study	Objective/Focus of previous study	Limitations	Significance of our study	Research gap
[9]	To provide knowledge on IoT security issues	The authors have not discussed any big data technologies	Primarily discusses the usage of big data technologies	Big data technologies
[13]	To survey on technologies and techniques for reliable and secure data communications	The authors have not discussed any big data technologies or about big data in general	Provides a detailed explanation on big data	Big data technologies
[14]	To facilitate the analytics and learning in IoT domain by providing overview of deep learning	An in-depth analysis has not been conducted with respect to IoT security. Further, IoT attack types have not been detailed	Discusses in detail on IoT security and its attack types	IoT security
[15]	To investigate state-of-the-art research in big IoT data analytics	Lacks in-depth study of IoT security	Performs an in-depth analysis with respect to IoT security	IoT security
[16]	To provide a comprehensive survey and taxonomy for existing security solution in vehicle-to-everything communication technology	The authors have not discussed big data technologies	The taxonomy details on big data technologies has been presented	Big data technologies
[10]	To discuss on major cybersecurity challenges and opportunities for cybersecurity + edge computing + IoT + Artificial Intelligence (AI)	The authors have not performed in-depth analysis of big data technologies	In-depth analysis has been conducted in the state-of-the-art big data technologies	Big data technologies
[2]	To address issues of real-time anomaly detection	Lacks in depth analysis of deep learning and attack types in the IoT space	Details on deep learning and IoT attack types	Deep Learning and IoT security
[17]	To provide comprehensive security analysis of IoT	Minimal discussion about deep learning and lacks discussion on big data technologies	Performs an in-depth discussion of deep learning and its algorithms, and also discusses big data technologies	Deep learning and big data technologies
[18]	To discuss on the most prominent attacks in IoT	The authors have not discussed deep learning and big data technologies	Discussion on deep learning and big data technologies has been presented	Deep learning and big data technologies
[19]	To discuss on various security challenges and threats with respect to their possible sources of occurrence	Lacks discussion on deep learning and big data technologies	Discussed about deep learning and big data technologies	Deep learning and big data technologies

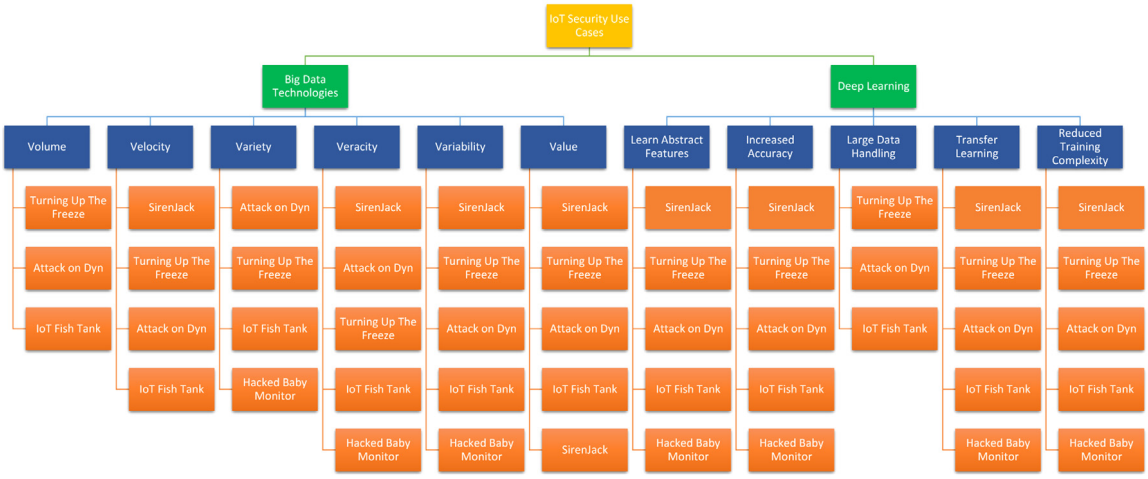

Fig. 1. IoT security use cases.

GB worth data was transmitted to a device located in Finland [37]. This use case provides us ample evidence that IoT devices can be used to manipulate an entire network. Hence, stopping cyber criminals at firewall is key to prevent any catastrophic incidents. Therefore, the continuous monitoring of data flow using big data technologies and deep learning would enable detection of IoT based security breaches at an early stage.

2.5. Hacked baby monitor

A baby monitor of a family in Ohio was hacked by an unknown hacker. When Adam and Heather Schreck and their 10-month old daughter were asleep, they heard a man screaming "Wake up baby! Wake up baby" from the baby monitor. When the baby monitor was inspected, the family found the camera angle moving on its own and the voice of the man screaming again. When Adam Schreck rushed into his daughter's room, the angle of the camera turned and pointed to his face and the man started screaming obscenities. The parents rushed to unplug the camera. Similarly, in Texas a family's wireless baby monitor was hacked and a similar wakeup call was heard from the baby monitor [38]. For hackers to get into a baby monitor, they have to use a network as the medium. This network can be secured

by combining deep learning and big data technologies to detect any anomalous data or intrusion in real-time.

The above discussed use cases are some of the sophisticated attacks on IoT. Nevertheless, these types of attacks on IoT are growing continuously and require modern day and most novel solutions. These complex attacks can be handled by deep learning due to its distinguishing features such as, capability of learning more abstract features, reduced training complexity of the model, promising accuracy, capability to handle large datasets, and support for transfer learning [39–42]. Additionally, big data technologies can play a vital role in processing of IoT data, especially due to the volume, velocity, and variety of data generated by IoT devices. Existing methodologies are inefficient in handling these types of data, thus big data technologies become a necessity [43]. Furthermore, big data technologies have also seen increased performance compared to traditional methods as illustrated by [44] where the training time is much lesser compared to the regular training method.

This section has discussed the motivation for this study and some of the recent real world attacks on IoT as use cases. Further, we have explained how deep learning and big data technologies can contribute to IoT security.

3. Background

This section contains a comprehensive description of deep learning, big data technologies, and IoT security. Additionally, the relationship between these three domains have been discussed, to provide fundamental knowledge and relationship mapping on these leading edge topics.

3.1. Deep learning

Deep Learning is a subset of machine learning which has three learning techniques, namely, supervised, semi-supervised and unsupervised learning. It consists of many layers of artificial neural networks. Each of the layer contains some neurons with activation functions that can be utilized to produce non-linear outputs. This methodology is said to be inspired by the neuron structure of the human brain [45,46].

In recent years, deep learning has attracted many researchers and organizations, compared to traditional machine learning approaches. The authors of [14] have compared deep learning against four machine learning algorithms, such as, Support Vector Machine (SVM), Decision Trees, K means, and Logistic Regression using Google trends, and the results indicate that deep learning is becoming more popular. Furthermore, this technology has been applied in a variety of AI applications such as, image recognition, image retrieval, search engines and information retrieval, and natural language processing.

Machine learning and deep learning have four phases in building a model. Fig. 2 illustrates the difference between machine learning and deep learning.

As discussed in Section 2, deep learning has gained recognition due to its characteristics of being capable of learning more abstract features, reduced training complexity of the model, promising accuracy, capability to handle large datasets, and support for transfer learning [39–42].

Deep learning in general has been explained in this subsection. Followed by the discussion of the typical methodology and characteristics of deep learning.

3.2. Big data technologies

Big data can be described as the high-volume, high-velocity, and high-variety of information that demands innovative forms of information processing to gain insights and for decision-making [47]. Typically, big data is characterized with 6 traits, generally referred to as the 6V's.

Fig. 3 illustrates the 6V's, which are the basic characteristics of big data, in general. However, data is classified as big data as long as it fulfils the first 3V's which are volume, velocity, variety [48]. Big data technologies can be described as the tools or technologies that are used to efficiently process data that has been classified as big data. Some of the big data technologies include, Apache Hadoop [49], Apache Spark [50], Apache Storm [51], Apache Flink [52], Apache Cassandra [53], and Apache HBase [54].

In the above section we had illustrated the characteristics of big data, which are the 6V's. Additionally, we had also listed some of the commonly used big data technologies.

3.3. IoT security

IoT enables sensors and devices in a smart environment to communicate with each other and enables information sharing across platforms. Recently IoT has been widely adopted into building intelligent systems such as, smart city, smart home, smart office, smart retail outlets, smart agriculture, smart water management, smart transportation, smart healthcare, and smart energy [15,55,56].

Due to the wide use of IoT in mobile devices, transportation facilities, public facilities, and home appliances, these equipment can be used for data acquisition in IoT. Furthermore, devices used in various applications that are connected to the IoT network can be controlled remotely. The devices can communicate with each other and also with the central controlling devices. Additionally, when deployed in various domains, variety of data can be collected such as, geographical, astronomical, environmental, and logistical data [15].

IoT security is regarded as securing the entire deployment architecture of IoT from attacks [57]. There are various factors that needs to be taken into consideration for developing IoT security solutions. The following are the security requirements that needs to be met for developing IoT security solutions. Due to the immense capabilities made available by deep learning and big data technologies, they can be utilized to identify a pool of security breaches related to the security requirements.

3.3.1. Confidentiality

Confidentiality enables information to be transmitted securely during all communications. When information is transmitted without authentication or encryption, adversaries are given the chance to violate the privacy of the owner [58,59]. Typically, big data technologies consist of secure transmission of data by using encryption methodologies, thus preventing data to be compromised by adversaries [60].

3.3.2. Integrity

The integrity of an IoT system may be compromised by an adversary. Therefore, integrity guarantees that data received has not been manipulated during transmission [59,61]. In addition, Apache Spark, a big data technology enables the support for data quality checks in the Spark DataFrame [62]. This enables users to perform data integrity checks on the IoT system.

3.3.3. Availability

Availability in IoT systems refer to ensuring that legitimate users are able to access the system and that unauthorized access is denied [59,63]. One of primary goals of big data technologies is to ensure its omnipresence to the user. Further, they can be run on multiple nodes that ensures high availability of the application [64].

3.3.4. Authentication

Authentication refers to ensure the identity of the peer which IoT devices communicate with. Furthermore, it is also concerned with valid users gaining appropriate access for network tasks such as control of IoT devices and networks [59,61]. Additionally, big data technologies such as Apache Spark incorporate authentication mechanisms for Remote Procedure Call (RPC) channels [60].

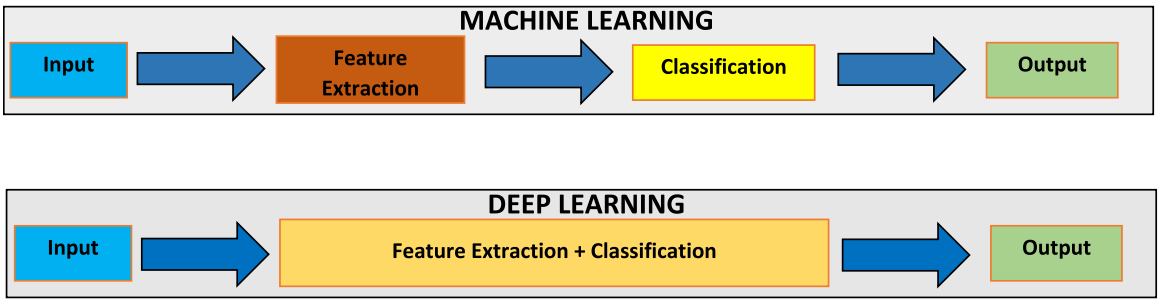

Fig. 2. Machine learning vs. deep learning.

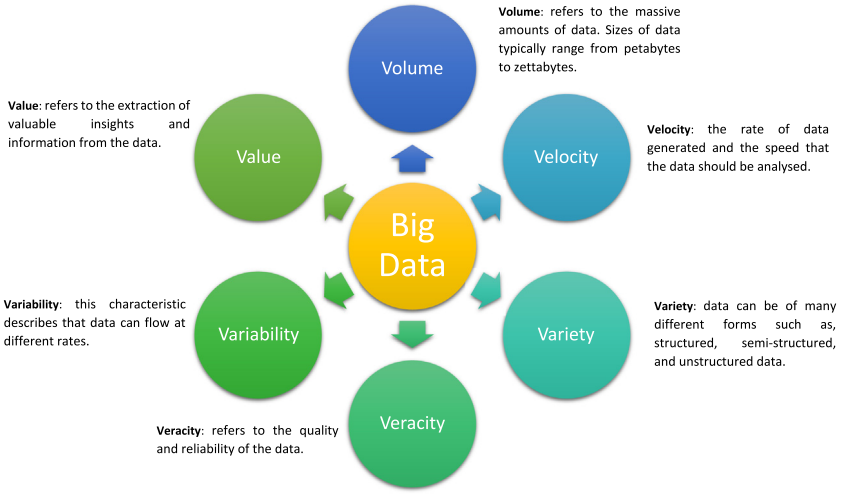

Fig. 3. 6V's of big data.

3.3.5. Access control

Access control in IoT system should act as a means of ensuring that the authenticated nodes are limited to access what they are privileged to and nothing more [59,61]. Furthermore, it is known that big data technologies provide access control support for its applications. A filter is necessary for this to be achieved and each application can be equipped with its own access control list [60]

Even though, deep learning is not directly related to the IoT security requirements, the continuous monitoring of networking and communications between the IoT devices and system can aid in detecting and mitigating security breaches at an early stage. As discussed in Section 3.1, the characteristics of deep learning contribute to the identification of security breaches, this is because deep learning is capable of handling very large datasets, classifying legitimate data and anomalous data at a higher accuracy rate, learning from complex data, and learning from data at a much faster pace.

Fig. 4 illustrates the connection to benefits of IoT devices.

The above sections had discussed on deep learning, big data technologies and IoT security along with the relationship between them. We have further elaborated the aforementioned topics in the following sections.

4. Taxonomy

This section highlights and proposes a taxonomy for deep learning, big data technologies, and IoT security. This taxonomy is classified into different categories namely, Deep Learning, IoT Security, and Big Data Technologies, and further sub categorized as Deep Learning Architectures, Frameworks, Model Evaluation, IoT Security Application Area, IoT Security Attacks, Datasets, Apache Hadoop, Apache Spark, and Apache Storm. Due to the limited studies that have been conducted by combining deep learning, big data technologies, and IoT security, we have identified the relationship among these three domains based on related experimental studies that have used deep learning with a combination of either IoT security, or big data technologies, and IoT security or big data technologies with security attack detection, which consists of identical attacks as of that in the IoT space. The taxonomy derived has been illustrated in Fig. 5.

4.1. Deep learning

In this subsection, we have detailed the common deep learning architectures, popular deep learning frameworks, and the evaluation methods used to evaluate deep learning based models.

4.1.1. Deep learning architectures

Deep learning architectures generally have three types of learning models, supervised learning, unsupervised learning, and semi-supervised learning. In a supervised learning the data used to train the architecture is fully labelled, whereas in the unsupervised learning, the data is not labelled and the architecture tries to come up with a structure by extracting useful information. In semi-supervised learning model a training dataset contains a mixture of labelled and unlabelled data, this type of learning is futile when extracting relevant features from the data is tedious [65]. Further, deep learning architectures can be categorized into two types, discriminative and generative. The discriminative model generally supports supervised learning methods, whereas the generative model supports unsupervised learning methods [14].

i. **Autoencoder (AE)**: AE is a type of Artificial Neural Network (ANN) that learns efficient data coding in an unsupervised fashion [66,67]. AEs comprise of an input and an output layer that

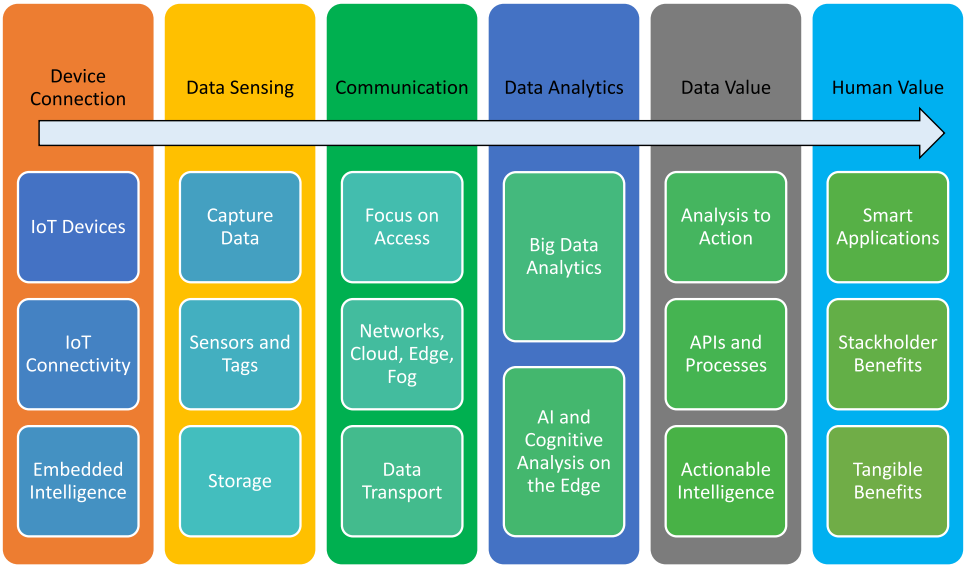

Fig. 4. Device connection to human value in IoT.

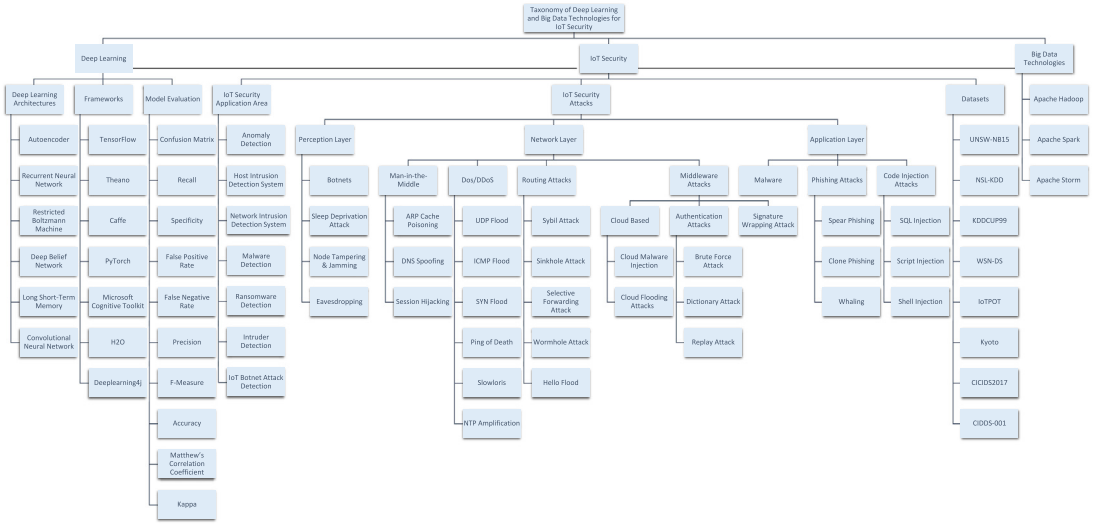

Fig. 5. Taxonomy of deep learning, big data technologies and IoT security.

are connected using one or more hidden layers. Generally, AEs consist of the same number of input and output layers. It aims in transforming inputs to outputs in the simplest way possible, by ensuring the input is not distorted very much [14].

ii. **Recurrent Neural Network (RNN)**: RNN are said to be an extension of Feed Forward Neural Network (FFNN), which takes advantage of sequential information. RNNs get the name recurrent as they perform the same task for each element of a sequence, where the output is dependent on previous computations [68].

iii. **Restricted Boltzmann Machine (RBM)**: RBM is a kind of ANN with the capability of representing and solving difficult problems. The RBM comprises of two process types, learning and testing. In the learning phase, vast amount of input examples and desired outputs are presented to generate the RBM structure where a general rule of mapping inputs to outputs is learned. In the testing phase, outputs are produced for new inputs by the RBM, abiding the general rule that was obtained in the learning phase [69].

iv. **Deep Belief Network (DBN)**: DBN is a type of Deep Neural Network (DNN) that comprises of multiple layers of hidden units, where there are connections between the layers but not with the units of each layer. Further, DBNs can learn to probabilistically reconstruct its inputs when trained with examples in unsupervised learning. Additionally, on the post learning phase a DBN can be trained further with supervised learning for classification problems [70–72].

v. **Long Short-Term Memory (LSTM)**: LSTM consists of special units often referred to as memory blocks in the recurrent hidden layer. Further, the memory blocks comprise of memory cells with self-connections storing the temporal state of the network in addition to the special multiplicative units referred to as gates, which controls the flow of information. Each memory block consists of input and output gates, where the input gate is responsible for the flow of input activations into the memory cell, and the output gate is responsible for output flow of cell activations into the rest of the network [73].

vi. **Convolutional Neural Network (CNN)**: CNN is a type of deep ANN which was first proposed by the authors of [74,75]. The CNN incorporates the back propagation algorithm for learning the receptive fields of simple units. Furthermore, the CNN is characterized by local connections, weight sharing and local

pooling properties. The local connections and weight sharing enable the model to discover local informative visual patterns with few adjustable parameters. The local pooling property equips the network with some translation invariance [76].

Table 2 classifies the architectures discussed above based on the category, learning model, and the studies that have utilized these architectures. Furthermore, the relationship and applicability of these architectures with big data technologies and IoT security have been proved by substantiating the success of the implementation.

4.1.2. Frameworks

The popular frameworks that are typically used for implementing deep learning architectures (see Section 4.1.1) are as follows.

i. **TensorFlow**: TensorFlow is an innovative framework developed by Google, which offers a variety of deep learning computation. TensorFlow was officially released in the late 2015. It includes Java, C++, Go and Python Application Programming Interface (API)s, and is primarily designed for computation on data flow graphs. Furthermore, TensorFlow supports multi-CPU and multi-GPU computations with CUDA and SYCL extensions. Additionally, TensorFlow Lite has been developed to provide support for mobile and embedded machine learning. Further, TensorFlow Lite provides an Android Neural Network API [90].
ii. **Theano**: Theano is an open source Python library, used for developing complex algorithms through mathematical expressions. It is typically utilized for machine learning researches. Furthermore, it has gained wide acceptance among the deep learning community due to its support for automatic symbolic differentiation and GPU accelerated computing. CUDA is used by Theano as one of its main backend for GPU accelerated computation [91].
iii. **Caffe**: Caffe is a widely used training infrastructure, developed by Berkeley Vision and Learning Center (BVLC) for deep learning based operations. DNNs are simulated as a network of computing units in Caffe. The computing units are generally referred to as "layers", these layers take data as input, perform a set of operations, and pass the output to the following layer [92].
iv. **PyTorch**: PyTorch is a deep learning framework based on python that acts as a replacement for NumPy to use the power of GPUs and for deep learning research that provides maximum flexibility and speed [93]. PyTorch is widely known for its two prominent features, strong GPU acceleration support and building neural networks dynamically [94].
v. **Microsoft Cognitive Toolkit (CNTK)**: CNTK is an open source deep learning framework for Windows and Linux. It is used for training and evaluating powerful deep neural networks. Microsoft uses this toolkit for Cortana speech models and web rankings. CNTK supports a variety of feed forward, convolutional, and recurrent networks for speech, image, and text data, and also a combination of these data. Furthermore, CNTK can scale to multiple GPU servers and is designed in aiming for efficiency [95].
vi. **H2O**: H2O is a fast, scalable, and open-source machine learning and deep learning framework for developing smart applications. The support for advanced algorithms such as deep learning, boosting and bagging elements make H2O preferable for smart applications. H2O is capable of handling billions of data row in-memory even in a small cluster. H2O is typically designed to start deploying within minutes and provides support for Apace Hadoop and Apache Spark cluster [96].
vii. **Deeplearning4j**: Deeplearning4j is an open source framework for deep learning computations developed by a team led by Adam Gibson and supported by the organization SkyMind. This framework was written in Java, Scala, CUDA, C, and C++ and is distributed under the Apache license 2.0. Furthermore, it is compatible with Linux, OS X, Windows, and Android. Deeplearning4j supports implementation of all deep nets such as, RBM, DBN, Deep Autoencoder (DAE), and more [97].

Table 3 describes some frameworks commonly used for deep learning, the programming languages they were written in, the latest stable release version, and the latest stable release date.

4.1.3. Model evaluation

The commonly used model evaluation techniques for deep learning based models are as follows.

i. **Confusion Matrix**: Confusion matrix is a summary of the predicted results of the classification model. The confusion matrix is derived by summarizing the total count of correctly and incorrectly classified predictions based on each class [98]. It is necessary to derive the following values before designing the confusion matrix:

   (a) **True Positive (TP)**: The true positive values refer to the number of instances that has been correctly classified by the model [99].
   (b) **True Negative (TN)**: The true negative values are the number of negative instances that were correctly classified by the model [99].
   (c) **False Positive (FP)**: False positive value is the number of negative instances labelled incorrectly as positive instances [99].
   (d) **False Negative (FN)**: False negative value is the number of positive instances labelled incorrectly as negative instances [99].

   Table 4 explains the confusion matrix.
ii. **Recall**: Recall also referred to as sensitivity or true positive rate refers to the proportion of real positive instances that have been predicted positive [100]. Recall can be calculated using the below formula.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

iii. **Specificity**: Specificity describes the effectiveness of the classification model in identifying negative labels [101]. Specificity is calculated using the below formula.

$$Specificity = \frac{TN}{TN + FP} \tag{2}$$

iv. **False Positive Rate (FPR)**: FPR also called the Fall-Out is the proportion on negative instances classified incorrectly as positive instances. In simpler terms, probability of false alarms to be raised [102]. The FPR is calculated using the below formula.

$$FalsePositiveRate = \frac{FP}{TN + FP} \tag{3}$$

v. **False Negative Rate (FNR)**: FNR refers to the proportion of incorrectly classified samples to the number of positive samples [103]. The FNR is calculated using the below formula.

$$FalseNegativeRate = \frac{FN}{TP + FN} \tag{4}$$

vi. **Precision**: Precision is the proportion of predicted positives that are real positives. Precision is applied on a variety of areas such as, machine learning, data mining, and information retrieval [100]. Precision is calculated using the below formula:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Table 2
Deep learning architectures.

Architectures	Category	Learning model	Studies
AE	Generative	Unsupervised	[23,77]
RNN	Discriminative	Supervised	[78]
RBM	Generative	Unsupervised & Supervised	[79–81]
DBN	Generative	Unsupervised & Supervised	[82,83]
LSTM	Discriminative	Supervised	[25,84–87]
CNN	Discriminative	Supervised	[86–89]

Table 3
Deep learning framework.

Framework	Written in	Latest stable release version	Latest stable release date
TensorFlow	Python, C++, and CUDA	1.12.0	5th November 2018
Theano	Python, and CUDA	1.0.4	16th January 2019
Caffe	C++	1	18th April 2017
PyTorch	Python, C++, and CUDA		7th February 2019
CNTK	C++	2.7	4th January 2019
H2O	Java	3.24.0.3	7th May 2019
Deeplearning4j	Java, Scala, CUDA, C, C++, Python, and Clojure	0.9.1	13th August 2017

Table 4
Confusion matrix.

	Actual positive	Actual negative
Predicted positive	TP	FP
Predicted negative	FN	TN

vii. **F-Measure**: The f-measure is said to be the harmonic mean of the precision and recall [82]. The f-measure is calculated using the below mathematical equation.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6}$$

viii. **Accuracy**: Accuracy can be described as the overall effectiveness of the classification model [101]. The formula used for the calculation of the accuracy is as follows:

$$AC = \frac{(TP + TN)}{TP + FP + TN + FN} \tag{7}$$

ix. **Matthew's Correlation Coefficient (MCC)**: MCC is a technique used for measuring the quality of binary and multiclass classification. The MCC values ranges from −1 to +1, where −1 denotes total disagreement, 0 indicates random predications and +1 indicates total agreement [104,105]. The MCC can be calculated using the below formula:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{8}$$

x. **Kappa**: Kappa also referred to as Cohen's Kappa is a measure of the inter-reliability. Kappa is said to be more robust compared to the simple percent agreement method. Kappa values range from 0–1, the following list is the interpretation of Kappa [106]:

   i. 0–0.20 No Agreement
   ii. 0.21–0.39 Slight Agreement
   iii. 0.40–0.59 Fair Agreement
   iv. 0.60–0.79 Substantial Agreement
   v. 0.80–0.90 Almost Perfect

Kappa is calculated using the below formula:

$$k \equiv \frac{P_o - P_e}{1 - P_e} = 1 - \frac{1 - P_o}{1 - P_e} \tag{9}$$

Table 5 highlights some studies that have incorporated the discussed model evaluation techniques to evaluate their models.

In this subsection, we detailed about common deep learning architectures along with popular deep learning frameworks. Finally, we discussed on the evaluation methods used for evaluating deep learning based models.

4.2. IoT security

This subsection will discuss on IoT security application areas where deep learning has prominently been applied with a focus on IoT, the security attack types on the IoT space where deep learning can be used to identify and mitigate those attacks, and finally the datasets that contain IoT based attacks.

4.2.1. IoT security application areas

The common IoT security application areas where deep learning has prominently been applied has been discussed below.

i. **Anomaly Detection**: Anomaly detection is the process of identifying anomalies. Anomalies are often referred to as patterns that do not follow a standard pattern. These anomalies are generated by abnormal activities such as, cyber-attacks, credit card frauds, and more. An anomaly is generally categorized into three categories, namely point anomalies, contextual anomalies, and collective anomalies.

   (a) **Point anomalies**: If a data instance differs from a normal pattern in the dataset, it is said to be a point anomaly.
   (b) **Contextual anomalies**: If in a particular context, the data instance behaves anomalously then it is called contextual anomalies.
   (c) **Collective anomalies**: If a group of similar data instances behaves anomalously compared with the entire dataset, they are said to be collective anomalies [113].

ii. **Host Intrusion Detection System (HIDS)**: HIDS are used to monitor activities and characteristics of a single host in a network for any abnormal activities. Generally, agents are deployed onto target hosts in host-based intrusion detection systems. In some cases, the agents may be deployed on remote devices. Sensors in host-based intrusion detection systems are deployed as inline or passive. In inline sensors, the network

Table 5
Deep learning model evaluation.

Study	Confusion matrix	Recall	Specificity	FPR	FNR	Precision	F-Measure	Acc.	MCC	Kappa
[89]		✓				✓	✓	✓		
[23]				✓				✓		
[24]		✓				✓	✓	✓		
[25]	✓	✓		✓		✓	✓			
[85]								✓		
[80]	✓	✓	✓	✓	✓	✓	✓	✓		
[45]		✓				✓		✓		
[86]		✓				✓		✓		
[107]				✓						
[87]		✓	✓	✓		✓	✓		✓	
[108]		✓	✓					✓		
[82]		✓				✓	✓			
[78]										✓
[109]				✓		✓	✓	✓		
[110]										
[77]				✓				✓		
[81]				✓	✓			✓		
[44]				✓	✓			✓		
[29]		✓	✓					✓		
[30]		✓				✓	✓	✓		
[111]	✓					✓	✓	✓	✓	
[112]								✓		

traffic passes through the sensors and then reaches the target hosts. The passive sensors monitor a replica of the real network traffic [114].

iii. **Network Intrusion Detection System (NIDS)**: A NIDS is used to monitor network traffic flow. The different network layers are analysed by NIDS to detect any possible security threats [114].

iv. **Malware Detection**: Malware detection is the process of identifying malware. Typically, there are two types of malware detection, which are static or dynamic analysis. In the static analysis, the malware is directly analysed in its binary form, whereas, in the dynamic analysis, the binary files are executed and the activities are monitored [115].

v. **Ransomware Detection**: A ransomware is a type of malware which encrypts the affected computer and a ransom is demanded for decryption [116]. Ransomware detection is the process of identifying ransomware attacks.

vi. **Intruder Detection**: Intruder detection is the process of identifying intruders with precise information. Intruders fall into the following 3 categories:

   (a) **Masquerader**: A person trying to gain unauthorized access into a system
   (b) **Misfeasor**: An authorized user who tries to access privilege features which the users is prohibited from accessing.
   (c) **Clandestine user**: A person who gains supervisory control of a system in order to evade auditing and access control or to suppress audit collection [78].

vii. **IoT Botnet Attack Detection**: A bot is a device connected to a common protocol infrastructure which is remotely controlled. A device can be compromised and turned into at bot by attackers. When an IoT device joins a botnet, the device can be utilized for a variety of purposes, including DDoS attacks [117]. IoT botnet attack detection is the act of detecting IoT botnet based attacks such as, DDoS.

Table 6 denotes the IoT security application areas where deep learning, mainly with big data technologies have been applied.

4.2.2. IoT security attacks

Various IoT security attack based on each layer are as follows.

4.2.2.1. Perception layer attacks. The perception layer consists of physical objects such as, sensors and actuators, nodes, and devices. A perception layer attack affects the physical object in the IoT infrastructure. Common perception layer attacks have been elaborated below.

i. **Botnets**: Botnets such as Mirai, comprises of four major components: (i) a bot is the malware which infects devices. The bot primarily aims in conducting two tasks, which is to infect misconfigured devices and to attack a target server on receiving the command from a botmaster, the person controlling the bot, (ii) a centralized management interface monitors the condition of botnet and orchestrates the attack provided to the botmaster through a Command & Control (C&C) server, (iii) the loader spreads the executables targeting various types of platforms such as, Acorn RISC Machine (ARM), MIPS, and x86, through direct communication with new targets, and (iv) the report server is used to maintain a list of devices in the botnet [23,118].

ii. **Sleep Deprivation Attack**: Sleep deprivation attack is a type of attack conducted on battery powered sensor nodes and devices. Typically, battery powered devices follow a sleep routine in order to extend its lifetime. The sleep deprivation attack aims in keeping the nodes and devices awake for an extended period of time, which results in more battery power consumption and eventually shutting down of the nodes and devices [119].

iii. **Node Tampering & Jamming**: Node tampering attacks are triggered when an entire node or part of the node's hardware is replaced physically. Electronically way node tampering can be achieved by interrogating the nodes to gain access and manipulate sensitive information, such as, routing tables, and shared cryptographic keys. Whereas, a node jamming attack is when an attacker interferes with the radio frequencies of wireless sensor nodes, which jams the signal and delays communication to the nodes. Provided that the attacker is able to jam key sensor nodes, IoT services can be denied [120].

iv. **Eavesdropping**: Eavesdropping is an attack that threatens the confidentiality of a message. An eavesdropping attack is when the attacker overhears information that is passed via a private communication channel. It is said that the Radio Frequency Identification (RFID) is the most susceptible device for eavesdropping kind of attacks [61].

4.2.2.2. Network layer attacks. The network layer generally consists of network components such as, routers, bridges, and other types

Table 6
IoT security application areas.

IoT security application area	Studies
Anomaly detection	[81,82]
HIDS	[30]
NIDS	[25,29,30,44,45,80]
Malware detection	[85,89,109]
Ransomware detection	[87]
Intruder detection	[78]
IoT botnet attack detection	[23]

of networking components. A network layer attack is an attack directed towards disrupting the network components in the IoT space. Prominent network layer attacks in IoT have been discussed below.

i. **Man-in-the-Middle (MIM)**: In the MIM attack an attacker has total control over a communication channel between two legitimate entities. Further, the attacker is not limited to reading messages, but to change, erase, and insert messages into the communication channel [121].

   (a) **Address Resolution Protocol (ARP) Cache Poisoning**: The ARP protocol targets the resolution of MAC addresses of a host given its IP. This is achieved by transmitting an ARP packet request on the network. ARP cache poisoning is also referred to as ARP spoofing, ARP poison routing is the process of counterfeiting ARP packets that enables impersonation of another host on the network [122].
   (b) **DNS Spoofing**: A DNS maps symbolic names to the IP address. A DNS spoofing sometimes referred to as DNS cache poisoning, impacts the DNS resolver by storing malicious mapping information between symbolic names and IP addresses. The DNS server may be poisoned by an attacker by compromising an authoritative DNS server or forging a response to a recursive DNS query [123].
   (c) **Session Hijacking**: A session hijacking attack is the malicious act of the attacker who manages to secure the user's session identifier, allowing the attacker to transfer the session to his/her own system [124].

ii. **Denial of Service (DoS)/DDoS**: DoS is a type of malicious attack that aims in consuming resources or bandwidth of genuine users. A DDoS is a variant of the DoS which is similar to the DoS attack but involves various compromised nodes. [125].

   (a) **User Datagram Protocol (UDP) Flood**: UDP flood is a flooding attack where multiple UDP datagrams are generated typically by a bot. These UDP datagrams flood through various parts of the network and congest the entire network [126].
   (b) **Internet Control Message Protocol (ICMP) Flood**: ICMP flood referred to as ping floods where a continuous ICMP Echo Request (ping) packets are sent to the host as fast as possible without waiting for a reply. This will consume incoming and outgoing communications resources as the host tries to reply to the pings [125].
   (c) **SYN Flood**: In a SYN flood attack an attacker sends vast amount of Transmission Control Protocol (TCP) SYN packets to a target. This forces the target to utilize constrained resources such as, CPU, bandwidth, and memory in order to reply to the SYNs. High velocity of attack will cause a DoS attack and eventually will be unable to serve genuine users [127].
   (d) **Ping of Death**: The ping of death is an attack, where an attacker sends an extremely large sized ping to the target with intention to collapsing the target. Many operating systems tend to crash when the ping size has been exceeded [128].
   (e) **Slowloris**: The Slowloris is a DDoS attack, where multiple HyperText Transfer Protocol (HTTP) requests are opened and manipulated simultaneously between the attacker and target. Slowloris are capable of collapsing an application by using minimal traffic and attackers [129].
   (f) **Network Time Protocol (NTP) Amplification**: NTP Amplification attack is a type of reflection-based volumetric DDoS attack where the NTP is exploited by the attacker to flood an amplified UDP traffic to a host. Hence, this affects the host and surrounding infrastructure causing regular traffic inaccessible to the resource [130].

iii. **Routing Attacks**: In routing attacks malicious nodes launch routing types of attacks to disrupt routing operation or for performing DoS attacks [131].

   (a) **Sybil Attack**: During Sybil attack a malicious node breaks the routing system, and accesses information blocked by the node, or the network gets partitioned. This attack is executed by a single attacker who creates multiple false identities and pretends to be multiple in peer-to-peer networks (P-2-P) [132].
   (b) **Sinkhole Attack**: Sinkhole attack is conducted by comprising a node which attempts to draw traffic as much as possible from a specific area, by making itself look appealing to the surrounding nodes based on the routing metric. Hence, the malicious node attracts all traffic from the base station. This then provides the attacker to conduct further attacks on the system [133].
   (c) **Selective Forwarding Attack**: A selective forwarding attack is capable of conducting a DoS attack where malicious nodes selectively forward packets. The goal of this attack generally is to disrupt routing paths. Nevertheless, it can be used to filter any protocol [134].
   (d) **Wormhole Attack**: The aim of a wormhole attack is to disrupt the network topology and traffic flow. The wormhole attack takes place when a malicious node tunnels messages among two different parts of the network through a high speed link [135,136].
   (e) **Hello Flood**: The hello flood is one of the main attacks in the network layer. The hello flood attack enables the attacker to force conventional nodes to lose power by forcing them to transmit large hello packets with very high power [137].

iv. **Middleware Attacks**: In the IoT infrastructure the middleware comprises of components such as cloud. A middleware attack directly involves malicious activities on the middleware components of the IoT infrastructure.

   (a) **Cloud Based**: In cloud based attack, the attackers directly attack a cloud platform for various reasons, such as information theft, flooding attack, and so forth. Common cloud based attacks include :

      i. **Cloud Malware Injection**: During cloud malware injection attack an attacker gains access to victim's

data in the cloud and uploads a malicious copy of the victim's service instance, therefore enabling the victim's service to be processed within the malicious instance [138].

ii. **Cloud Flooding Attack**: The cloud flooding attack enables the attacker send a huge number of packets from innocent host in the network in order to flood the victim. These huge packets can be a combination or multiple TCP, UDP, and ICMP. Furthermore, this type of attack can affect the service's ability to serve the authorized users. Additionally, the usage of the cloud may rise since it does not have the capability of identifying legitimate and attack traffic [139].

(b) **Authentication Attacks**: Authentication based attacks are used to exploit the authentication process that is used to verify a user, service, or application [140].

i. **Brute Force**: A brute-force attack makes an attacker to gain access by entering a variety of login credentials with the hope of guessing the credentials correctly. The attacker enters a variety of possible passwords until the correct password is found [141].

ii. **Dictionary Attack**: The dictionary attack also referred to as the password-guessing attack is when an attacker has built a database with possible passwords. The attacker executes this by eavesdropping on the channel and records the transcript. After that, passwords are attempted to be generated to match the recorded ones. If a match has been found, then the attacker has successfully managed to acquire the password [142].

iii. **Replay Attack**: A replay attack enables an attacker to intercept and capture a digital communication or action and use it at a later point of time. This enables the attacker to use someone else's information to masquerade as that person [143].

(c) **Signature Wrapping Attack**: Signature wrapping attack enables an attacker to appear as a legitimate user and perform arbitrary web service request. This is achieved by injecting a malicious element into the message structure, this ensures a valid signature for the legitimate elements and processing of the malicious element in the application logic [144].

4.2.2.3. Application layer attacks. An application layer is the application itself, such as smart homes, smart cities, and smart grids. An application layer attack is related to the security breaches of IoT applications. Prominent application layer attacks have been briefed below.

(a) **Malware**: Malware is a type of attack, where executable codes are used by attackers to disrupt the devices in the network. This enables the attackers to gain unauthorized access or steal sensitive information. In the IoT network, the attackers may take advantage of firmware flaws and are capable of disrupting the entire IoT architecture [145,146].

(b) **Phishing Attack**: Phishing is a type of attack which aims to extract sensitive information such as, usernames, and passwords from users by appearing to be a trustworthy entity. The sensitive information can be used later by cyber criminals to cause harm to the user or system [147].

i. **Spear Phishing**: Spear phishing is targeted specially at selected individuals and organizations, rather than random users. The attacker generally enhances his knowledge on the target and settings. The attacker then may send a message pretending to be of a legitimate entity [148].

ii. **Clone Phishing**: Clone phishing is when a legitimate email that was sent previously is cloned into a malicious email which generally contains a link to the phisher's website [148].

iii. **Whaling**: Whaling is similar to spear phishing except that it is mainly targeted at senior corporate executives and government officials [148].

(c) **Code Injection Attack**: A code injection attack focuses on depositing malicious executable code (machine code) into the address space of the victim's process, and then authorizes control over to this code [149].

i. **Structured Query Language (SQL) Injection**: SQL injection executes malicious SQL database statements by taking advantage of the insufficient validation of data flow from the user to the database [150].

ii. **Script Injection**: During script injection or Cross-Site Scripting (XSS) a malicious script, generally written in JavaScript is injected into the content of the website. The malicious script is capable of leaking sensitive information from the site [151].

iii. **Shell Injection**: Shell injection attacks sometimes referred to as command injection attacks inject malicious commands into a system to perform malicious activities [152].

Table 7 details on some of the notable attacks on the IoT space for each layer.

4.2.3. Datasets

The prominently used datasets for experimental analysis on deep learning, big data technologies and/or for IoT security or network security are as follows.

i. **UNSW-NB15**: The UNSW-NB15 dataset was developed in 2015, which consists of a combination of real modern normal and contemporary synthesized attack data. This is a labelled dataset and consists of a total of 47 features. Further, this dataset consists of 9 attack types, namely fuzzes, analysis, backdoors, DoS, exploits, generic, reconnaissance shellcode, and worm attack types [157].

ii. **NSL-KDD**: This dataset is an extension of the KDDCUP99 dataset, where selected records are extracted from the entire KDDCUP99 dataset. In study [158], the authors have asserted that the KDDCUP99 dataset highly affects performance of evaluated systems and results in poor evaluation of anomaly detection techniques. Therefore, they have proposed the NSL-KDD, which excludes redundant records in the train set, the proposed test sets do not contain duplicate records, on the hand in each difficulty level the number of records selected are inversely proportional to the percentage of records in the KDDCUP99 dataset, the train and test sets records are reasonable. NSL-KDD dataset comprises of four attack types, namely DoS, User to Root (U2R), Remote to Local (R2L), and Probe attacks.

iii. **KDDCUP99**: The KDDCUP99 dataset was created by the authors of study [159] based on the DARPA'98 IDS evaluation program [160]. Additionally, this dataset is widely used among researchers for the evaluation of anomaly detection approaches. The DARPA'98 dataset is about 4 GB of tcpdump data of 7 weeks of network traffic. Further, the training data of the dataset consists of approximately 4,900,000 single vector connections in which each consists of 41 features, labelled as attack or normal data. This dataset comprises of 4 types of attacks, DoS, U2R, R2L, and Probe attacks [158].

Table 7
Notable attacks on IoT.

Attack layer	Attack type	Year	Title	Description	Results/Impact	Citation
Physical	Botnet	2012	Carna botnet	Used to measure the extent of Internet	Carna found total 1.3 billion internet protocol version 4 (IPv4) addresses in use, where 141 million were behind a firewall and 729 million reverse DNS records. Remaining 2.3 billion IPv4 address was not used	[153,154]
Network	DNS spoofing/DNS hijacking	2017	–	Used DNS hijacking to attack 40 government agencies, telecom companies, and internet titans across 13 countries for 2 years	Update DNS records of organization so information will be routed to hackers defined servers	[155]
Application	Malware	2016	Mirai	Attack on Dyn DNS service provider. Mirai malware installed on large number of IoT devices	High-profile websites inaccessible such as Twitter, The New York Times for approximately 5 h in the United States	[156]

iv. **WSN-DS**: WSN-DS dataset was created by [161] based on network traffic in wireless sensor nodes. This dataset consists of a total of 26 labelled features. Additionally, the WSN-DS consists of 4 types of DoS based attacks, namely black hole attacks, grayhole attacks, flooding attacks, and scheduling attacks [161].
v. **IoTPOT**: IoTPOT dataset was developed by [162] which consists of IoT network traffic. This dataset consists of normal and malware based network traffic, primarily used in DDoS based attacks. The dataset is classified based on 5 malware families, namely ZORRO, GAYFGT, nttpd, KOS, and *.sh [162].
vi. **Kyoto**: Kyoto dataset was built in 2006 for Intrusion Detection System (IDS) research. This dataset is built based on 3 years of real network traffic data. Further, 14 features derived from the KDDCUP99 and as well as additional 10 features have been included in this dataset. Further, their honeypot data consists of a total of 50,033,015 normal sessions and 43,043,255 attack sessions. In addition, it is discussed on the 3 attack types, exploits, shellcodes, and malware [163].
vii. **CICIDS2017**: The CICIDS2017 dataset was created by the Canadian Institute for Cybersecurity (CIC) in 2017. It contains real world benign and attack network traffic data. This dataset consists of 225,746 records with a total of 80 features. Additionally, this dataset consists of Brute Force, Web, DoS, Botnet, and DDoS types of attacks [164].
viii. **Coburg Intrusion Detection Data Sets (CIDDS)-001**: CIDDS-001 is a labelled flow based dataset developed for anomaly based NIDS evaluation. The dataset consists of normal and attack traffic data collected over the period of four weeks. Further, this dataset consists of 14 features and 4 types of attacks such as , DoS, PortScan, Brute Force, and Ping Scan [165].

Table 8 describes the attack types that each dataset contains and list the studies that have used the dataset for experimental analysis on deep learning, big data technologies and/or for IoT security or network security.

An in depth discussion of the IoT security application areas, security based attacks on IoT on each layer, and datasets used for deep learning based experimental analysis, has been presented in this subsection.

4.3. Big data technologies

This subsection discusses the existing noteworthy big data technologies implemented in the context of deep learning for IoT security or network security. Furthermore, the big data technologies, their development platform, their latest stable versions, their latest stable release dates, and some studies that have applied big data technologies with either deep learning and/or for IoT security or network security have been tabulated in Table 9.

4.3.1. Apache Hadoop

Apache Hadoop is a batch processing tool that provides scalability and fault-tolerance. Hadoop supports petabytes of data and enables applications to be run on multiple nodes. Furthermore, the log data is broken down into blocks and is sent to the nodes in the Hadoop cluster. Additionally, Hadoop is popular due to its capability of quick retrieval, searching log data, scalability, faster insertion of data, and fault tolerance [167].

4.3.2. Apache Spark

Apache Spark was developed as a unified model for distributed data processing by University of California, Berkely in 2009. Spark extends the MapReduce model with data sharing abstraction called as Resilient Distributed Dataset (RDD). Using this extension, the Spark can capture and process workloads such as, SQL, streaming, machine learning, and graph processing [50].

4.3.3. Apache Storm

Apache Storm is an open source real-time computation system. Storm enables convenient processing of streams of data in real-time. Further, it is capable of processing million tuples per second per node. Storm is fast, scalable, fault-tolerant, and user friendly. Moreover, storm provides capabilities to incorporate databases in the processing [51].

Table 9 details the development platform of big data technologies, the latest stable release version,latest stable release date, and some studies that have applied big data technologies with either deep learning and IoT security or security attack detection.

Here, we discussed some of the prominently used big data technologies in the context of deep learning and IoT security.

5. State of the art deep learning for IoT security using big data technologies

This section comprises of three subsections. The first subsection presents insights of the state-of-the-art techniques in cases where deep learning has been applied for IoT security. The second subsection details on the application of deep learning along with big data technologies. Finally, a comprehensive review of deep learning, big data technologies and IoT security has been presented.

5.1. Deep learning and IoT security

This sub section discusses the state-of-the-art techniques used for IoT security using deep learning techniques. The IoT has gained so much attention that even the military use IoT. Internet of Battlefield Things (IoBT) is referred to as the usage of IoT for military operations and defensive applications. The authors of study [89] have identified that injection of malware is the most common attack. Further, they have proposed a deep Eigenspace learning approach to detect IoBT

Table 8
IoT based attack datasets.

Attacks	UNSW-NB15	NSL-KDD	KDD CUP99	WSN-DS	IoTPOT	Kyoto	CICIDS 2017	CIDDS-001
Normal	✓	✓	✓	✓	✓	✓	✓	✓
DoS	✓	✓	✓	✓			✓	✓
Probe		✓	✓					
R2L		✓	✓					
U2R		✓	✓					
Fuzzers	✓							
Analysis	✓							
Backdoors	✓							
Exploits	✓					✓		
Generic	✓							
Reconnaissance	✓							
Shell code	✓					✓		
Worms	✓							
Web							✓	
Botnet							✓	
DDoS							✓	
Malware					✓	✓		
PortScan								✓
BruteForce							✓	✓
Infilteration								
PingScan								✓
Studies	[25,30,45]	[24,30,45]	[30,80]	[30]	[23]	[30]	[30,86]	[166]

Table 9
Commonly used big data technologies.

Big data technologies	Development platform	Latest stable release version	Latest stable release date	Studies
Apache Hadoop	Java	3.1.1	August 8th 2018	[30,44,81,82,109,110]
Apache Spark	Scala, Java, Python, and R	2.4	April 23rd 2019	[29,30,77,78,81,82,109,110]
Apache Storm	Clojure and Java	1.2.2	May 17th 2018	[44]

malware through the device Operational Codes (OpCode) sequences. The OpCodes are transmuted into the vector space and deep Eigenspace learning approach is used to classify benign and malicious application. Additionally, they have evaluated the sustainability of the proposed approach against junk code insertion attacks. They have evaluated their model based on four evaluation metrics, namely accuracy, precision, recall, and f-measure. In addition, they have compared two other similar studies based on the metric. Comparatively, their proposed approach has achieved better accuracy of 99.68%, precision of 98.59%, recall of 98.37%, and f-measure of 98.48%. Further, the proposed model has been able to mitigate junk code insertion attacks [89]. Nevertheless, the datasets used in this study is a self-created dataset. The quality and validity of the data is debatable. Furthermore, a limited number of malware samples are included in the dataset.

IoT devices which are more easily compromised compared to desktop computers has led to a rise in IoT botnet attacks. In order to mitigate this threat, the authors of [23] have proposed the use of DAE to detect anomalous network traffic from compromised IoT devices. Deep learning has been applied on the extracted behaviour snapshot of the network. To evaluate their model, they have infected nine commercial IoT devices with the Mirai and BASHLITE botnets. The model was evaluated based on the True Positive Rate (TPR), False Positive Rate (FPR), and attack detection time. The TPR results received was 100%, while the mean of the FPR was 0.007 ± 0.01 for their proposed model. Furthermore, their model took 174 ± 212 ms to detect the attacks. However, the model has only been evaluated based on two botnets, namely the Mirai botnet and the BASHLITE botnet. Additionally, the proposed model has only been compared with three machine learning models. Comparison of other deep learning models will further clarify on the accuracy of the model.

Deep learning with its capabilities such as, high-level feature extraction capability, self-taught, and compression capabilities makes it an ideal hidden pattern discovery that aids in discriminating attacks from benign traffic. Therefore, study [24] proposes a deep learning approach based on Stochastic Gradient Descent (SGD), which enables the detection of attacks in the social IoT. The model has been evaluated based on accuracy, precision, recall, f1-measure, detection rate, and False Alarm Rate (FAR). The result show that deep models have outperformed shallow models in every evaluation aspect. Additionally, it is discussed that deep learning exhibits better performance compared to traditional machine learning models. In contrast, the attacks evaluated has been limited, such as DoS, Probe, R2L, and U2R attacks. Likewise, only a single dataset has been used to evaluate the model, the NSL-KDD dataset.

Besides, the authors of study [25] have proposed a deep learning technique that enables intrusion detection in IoT networks using the Bi-directional LSTM Recurrent Neural Network (BLSTM RNN). The model has been evaluated using seven metrics, namely accuracy, precision, recall, f1-score, miscalculation rate, FAR, and detection time. The proposed model was able to achieve a high accuracy of 95.7% . On the other hand, the proposed model has been evaluated on a single dataset. Also, the model was not compared with similar models in terms of evaluation.

Furthermore, in study [85] the authors have proposed a deep learning model using LSTM to detect malware in IoT based on OpCodes sequence. The model has been evaluated based on accuracy, TP, FP, TN, and FN. The accuracy acquired was 98% on new malware, malware not in the training data. On the contrary, the emulated dataset has been used in this study. Additionally, there has been limited dataset samples/files, with a total of 180 malwares and 271 benign files.

Additionally, authors of study [80] have introduced a framework for IoT based on Software Defined Networking (SDN). They primarily focused on IoT applications, where security is critical, like smart cities. They have utilized the RBM to deploy an IDS to detect anomalies. They have compared their proposed approach with machine learning algorithms and have evaluated them based on eight metrics, TP, FP, TN, FN, precision, recall, False Discovery Rate (FDR), and False Negative Rate (FNR). They were able to achieve a precision rate of more than 94%. Nevertheless, they have opted for the KDD99 dataset, this is an outdated dataset that contains attacks of the year 1999. Including recent datasets that contain modern day attacks will enhance the reliability of the model. Due to the fact that this dataset is outdated, they

only contain limited attack types such as, DoS, Probe, Reconnaissance, R2L, and U2R.

In study [45] the authors have discussed that IoT applications face major security issues in confidentiality, integrity, privacy, and availability. Therefore, they have proposed a model for cyberattack detection in the IoT environment. A total of four evaluation metrics have been used for model evaluation, which includes accuracy, precision, recall, and detection time. Their results revealed the robustness of the accuracy and significant time saving. However, the accuracy of the models have been above 95% for the NSL-KDD dataset whereas for the UNSW-NB15 all models have achieved an accuracy less than 95%. Further, the time also increases in the UNSW-NB15 compared to the NSL-KDD dataset. NSL-KDD is an extension of the KDD99 dataset with certain modifications made. Whereas, UNSW-NB15 dataset is a more recent dataset containing modern day attacks. It can be seen from the result that the model performs better on older datasets and performance decreases on recent datasets.

Additionally, the authors in study [86] have proposed and implemented a four deep learning algorithms and compared it against traditional machine learning algorithms. Further, they have identified that the hybrid LSTM + CNN algorithm have outperformed all other algorithms compared to deep learning and machine learning algorithms, with an astonishing accuracy of 97.16%. Comparatively, all deep learning models have outperformed the machine learning models. In contrast, the dataset was manipulated to balance the data as it consists of highly unbalanced data. In addition, limited model evaluation metrics were used such as, accuracy, precision, and recall. Further, evaluation metrics such as, f-measure, MCC, and TPR, may be of added value to the model.

Besides, the authors of study [107] have proposed a deep learning approach with Dense Random Neural Network (DRNN) to predict the probability of an ongoing network attack based on the packet capture. Their methodology primarily focuses on online detection of network attacks against IoT gateways. They have found that the results they obtained are comparable to those of the results from the simple threshold detector. Nevertheless, their study only focuses on limited types of attacks in the IoT space such as, UDP flood, TCP SYN, sleep deprivation attack, barrage attack, and broadcast attack. Further, the results have not been compared with other algorithms or with similar studies.

Ransomware, is a fast growing malware that has affected various industries in various countries. Therefore, study [87] proposes a model that uses LSTM and CNN to distinguish ransomware and goodware in networks. The evaluation metrics used for the model were f-measure, TPR, FPR, and MCC. It is claimed that the model acquires an f-measure of 99.6% with a TPR of 97.2% in the classification of ransomware. It is also described that the model has been able to identify new ransomware in a timely and accurate fashion. However, the study used an emulated dataset. Additionally, the model only works in identifying ransomware, not other types of network based attacks such as DoS attacks.

Table 10 discusses on the application area, deep learning algorithms, limitation of the study and the citation of the discussed state of the art for deep learning and IoT security.

The above discussed studies have implemented deep learning architectures for IoT security and have evidently proven that deep learning can be applied for IoT security. However, these studies have their own limitations that needs to be addressed in future studies.

5.2. Deep learning and big data technologies

This sub section discusses the state-of-the-art techniques used for deep learning and big data technologies. With the vast amounts of data generated by various industries, leads to the interest in developing big data tools for analysis. Thus, authors of study [108] have proposed a framework that incorporates Apache Spark and a Multi-Layer Perceptron (MLP) using cascade learning. There framework composes of three stages, first stage is the input of dataset into Apache Spark, second stage is the cascade learning method, and in the third stage deep learning algorithm is applied. The framework has been evaluated based on two metrics, f1 score and accuracy. They have claimed that they have been able to obtain a model that conducts large scale big data analysis within short periods of time, with lesser computational complexity and with significant higher accuracy. Needless to say, the accuracy and f1 score of the proposed model does not reach even 75% for all the stages. Furthermore, limited big data technologies have been incorporated into the proposed framework.

Besides, in study [82] the authors have strongly claimed that machine learning techniques are not robust enough to detect sophisticated attacks in existing IDS. Therefore, they have proposed a distributed approach for abnormal behaviour detection in large scale networks. They have used the DBN, multi-layer ensemble SVM, and Apache Spark to achieve their model. Their model has been evaluated using Area under Receiver Operating Characteristic (ROC), precision, recall, f-measure and training time. The model has shown high performance in detection of abnormal behaviour in a distributed way. Further, this model addresses the feature engineering step for ensemble learning, especially with large datasets. However, the training time for their proposed approach has been significantly higher compared to the other models they have evaluated. Further, the number of features in the dataset make an impact on the accuracy of the model.

Additionally, the authors in study [168] have designed and implemented a framework that trains DNN using Apache Spark. Training of deep learning models requires large amounts of data and is computational extensive. They have claimed that their proposed framework can accelerate the training time by distributing the model replicas, through the stochastic gradient descent, among nodes for data in Hadoop Distributed File System (HDFS). The framework was evaluated based on run time, accuracy, and error rate. The proposed framework has shown satisfactory performance of time and accuracy. In contrast, the run time of the model shows an increase when there is lesser number of nodes. Moreover, it is seen that the error rate decreases only as the number of iterations increase.

Furthermore, study [78] has proposed a framework to perform intruder detection and analysis using RNN and rule association mining. The framework employs Apache Spark for training after the dataset is normalized. The framework has been evaluated using the amount of correctly classified instances, incorrectly classified instances, Kappa, mean absolute error, root mean squared error, relative absolute error, and root relative squared error. the study was able to achieve 199 correctly classified instances (100%) and 0 incorrectly classified instances (0%). Further, a Kappa score of 1 has been achieved. On the other hand, the study had limited the model for intruder detection only. In addition, other evaluation metrics have not been considered for the model, such as training time, precision, and recall.

Netflow, a protocol used for network auditing analysis, and monitoring can be a source of information for incident detection and forensic purposes. Therefore, study [109] has proposed a method that incorporates NetFlows with Extreme Learning Machine (ELM) classifier, trained in a distributed environment of Apache Spark for malware activity detection. The model has been evaluated based on TPR, FPR, precision, accuracy, error rate, and f-measure. The proposed model yields higher accuracy, less error rate, and higher f-measure in most of the scenarios. However, in certain scenarios the method is executed the accuracy is deemed as the second highest compared to other models evaluated. Further, various big data technologies have not been considered.

Besides, the authors of the study [110] have proposed a DDoS detection method that uses neural networks, implemented on the Apache Spark cluster. By applying the Hadoop HDFS for its capability of creating fault-tolerant applications and efficiency in handling of large datasets, combined with neural networks, they were able to achieve an accuracy of 94%. They have affirmed that their system is capable of handling high velocity, and high volume network flow in real-time and is capable of distinguishing between genuine and attack data. Further,

Table 10
Deep learning and IoT security.

Application area	Deep learning architecture/model	Limitation of the study	Citation
Malware detection	Convolutional network	• Self-created dataset • Limited malware samples in dataset	[89]
IoT botnet attack detection	DAE	• Model evaluated only on Mirai and BASHLITE botnet • Proposed model compared with 3 machine learning algorithms	[23]
Attack detection	SGD	• Limited to DoS, Probe, R2L, and U2R attacks • Evaluated on a single dataset, NSL-KDD	[24]
Intrusion detection	LSTM + Bi-directional Recurrent Neural Network (BRNN)	• Evaluated on a single dataset • Model not compared with similar models	[25]
Malware detection	LSTM + Bidirectional Neural Networks (BNN)	• Emulated dataset • Limited dataset samples/files, 180 malwares and 271 benign files	[85]
Intrusion detection	RBM	• Outdated dataset, KDD99 used • Limited attack types, DoS, Probe, Reconnaissance, R2L, and U2R	[80]
Intrusion detection	Deep Feed Forward Neural Network (DFNN) + backpropagation	• NSL-KDD dataset accuracy above 95%, but accuracy drops on the UNSW-NB15 dataset, all < 95% • Time has a significant increase in the UNSW-NB15 dataset	[45]
IoT network cybersecurity	CNN + LSTM	• Manipulated dataset to become a balanced dataset • Limited model evaluation metrics, Accuracy, precision, and recall only	[86]
DDoS attack detection	DRNN	• Limited network attacks discussed, UDP flood, TCP SYN, sleep deprivation attack, barrage attack, and broadcast attack • No comparative analysis of similar studies for evaluation of model	[107]
Ransomware detection	LSTM + CNN	• Emulated dataset • Detection of ransomware only	[87]

they have claimed that Apache Spark is suitable for processing of large volume network traffic. Nevertheless, the accuracy can be further nourished using different deep learning algorithms or by incorporating optimization methods. Also, the model is only applicable to detect a single attack type.

Additionally, in study [77] the authors have proposed a system that incorporates two approaches, namely the anomaly-based distributed ANN, and signature-based approach. For the anomaly-based detector, BigDL deep learning library was used over Apache Spark. For the signature-based approach, Suricata an open source IDS was used. Their models have been evaluated based on FPR, accuracy, and DR. Their hybrid model has outperformed the traditional signature-based detector, and neural based anomaly-detector. On the contrary, limited metrics have been used to evaluate the model. Likewise, the model is only limited to detect a single type of attack.

Further, the authors of study [81] have proposed an anomaly detection method that uses the RBM and RNN for anomaly detection in power grids. The authors have primarily used Apache Hadoop and Apache Spark for analysing the heterogeneous data sources in power big data, and to apply their deep learning framework. Their model has been evaluated based on accuracy, FPR, and FNR. They were able to achieve high accuracy rates, low FPR, and low FNR. However, the model has not been trained on the benchmark datasets. In addition, limited evaluation metrics have been used for model evaluation.

Besides, in study [44] the authors have discussed on a framework for real-time intrusion detection. They have used a CC4 neural network which was proposed in study [169] and the MLP. Further, they have used Apache Storm to stream the data for real-time processing. They have asserted that the training time sees a significant reduction when using Apache Storm compared to the regular methods. They have evaluated the model based on accuracy, FPR, training time, and FNR. They have achieved 89% accuracy and 4.32% FPR. Nevertheless, the average accuracy falls below 90%, which can be further improved by incorporating other deep learning algorithms. In addition, the experiments have been conducted only on a single dataset.

Table 11 describes the big data technologies, deep learning architectures, application area, limitations, of the studies that have incorporated deep learning and big data technologies.

The discussed studies have utilized deep learning architectures and big data technologies primarily for security and have shown implementation success. Nevertheless, these studies have some limitations that can be overcome in future studies.

5.3. Deep learning and big data technologies for IoT security

This sub section discusses the relationship among the three prominent areas of our study. Further, we have elaborated on the state-of-the-art techniques for deep learning, big data technologies and IoT security. Additionally, we have tabulated the combination of the studies used in the state-of-the-art and identified the use of deep learning, big data technologies, and IoT security in each of these studies. Finally, we have discussed some of the prominently used cloud infrastructure that supports deep learning, big data technologies, and IoT security.

Based on our critical analysis we have made effort to address the relationship between deep learning, big data, and IoT security. However, past studies have only incorporated either deep learning and IoT security or deep learning and big data technologies. Furthermore, minimal study has been conducted on deep learning, big data technologies, and IoT security. This clearly makes it evident that there is a niche area for future researchers to address. Moreover, with our maximum effort of critically analysing a variety of studies, we have been able to identify only two studies that have discussed on all the three components. The advantages and shortcomings of the two studies have been described below.

Due to the exponential growth of various interconnected devices, innovative attacks have being conducted on these devices. Therefore, it is necessary to come up with innovative and fool proof methodologies to prevent catastrophic incidents. Hence, authors of [29] have designed a big data framework for intrusion detection using classification

Table 11
Deep learning and big data technologies.

Big data technologies	Deep learning architecture/Model	Application area	Limitation of the study	Citation
Apache Spark	MLP	Healthcare & tourism	• Accuracy and f1 score < 75% • Limited big data technologies used	[108]
Apache Spark	DBN	Network abnormal behaviour detection	• The training time is high compared to other models • Number of features in the dataset has an impact on the accuracy	[82]
Apache Spark	DNN	Big data applications	• Lesser the nodes, increase of run time • The error rate decrease only as iterations increase	[168]
Apache Spark	RNN	Intruder detection	• Limited to intruder detection • Other metrics not considered such as training time, precision, and recall	[78]
Apache Spark	ELM	Malware detection	• In scenario 1, the accuracy is second highest compared to other algorithms • Other big data technologies not explored	[109]
Apache Spark	ANN	DDoS attack detection	• Accuracy can be improved using other deep learning models • Limited to single attack	[110]
Apache Spark	AE	DDoS attack detection	• Other metrics not discussed such as training time • Limited to a single attack	[77]
Apache Hadoop Apache Spark	DRBM	Anomaly detection	• Model does not use benchmark datasets • Other metrics not discussed such as training time	[81]
Apache Storm	CC4 neural network + MLP	Real-time intrusion detection	• Average accuracy falls below 90% • Experimented on a single dataset	[44]

methods such as, DNN, SVM, random forest, decision tress, and naïve Bayes. The metrics used for evaluation are accuracy, recall, false rate, specificity, and prediction time. Apache Spark has been used as a platform for implementing intrusion detection in smart grids using big data analytics. They have claimed that the DNN algorithm gets the highest accuracy for the raw dataset. Nevertheless, the highest accuracy gained was by the DNN model, but the accuracy is less than 80%. Additionally, the DNN prediction time is higher compared to other models.

Besides, the authors in this study [30] have discussed the advancements in hardware, software, and network topologies, including the IoT, pose security threats that require modern day approaches to be implemented. Thus, they have proposed a DNN based IDS. The DNN used is MLP along with FFNN. It has been discussed that the framework has been developed based on big data technologies, Apache Spark cluster computing platform. The Apache Spark cluster computing is set up over the Apache Hadoop Yet Another Resource Negotiator (YARN). They have evaluated their model based on accuracy, precision, recall, f-score, TPR, and FPR. Moreover, their model has outperformed all other traditional machine learning approaches in both HIDS and NIDS. However, in the multi-class calcification the accuracy drops below 90% for certain attacks in some of the datasets. Further, the DNN's were not trained on the benchmark IDS datasets

Table 12 compares studies based on the inclusion of deep learning, big data technologies, and IoT security.

As seen from Table 12, only deep learning and IoT security or deep learning and big data technologies have been incorporated in these studies. Implementation success of studies [29] and [30], convinces researchers that deep learning and big data technologies can be combined for IoT security. Therefore, due to the limited study conducted on these areas, we encourage future researchers to implement models based on a variety of deep learning algorithms, and big data technologies for IoT security.

5.4. Cloud infrastructure for deep learning, big data technologies, and IoT security

This subsection details the cloud infrastructures that can be applied to deep learning, big data technologies, and IoT security. Deep learning has shown promising results in many domains, however deep learning maybe quite computational extensive on large scale applications. This in turn, forces the inclusion of additional computational resources. When deep learning is applied on a massive scale application, existing resource may be limited. Hence, cloud infrastructure can be utilized to solve this challenge as they contain vast amounts of resources such as, multi-core CPU, multi-core GPU, memory, and bandwidth. Additionally, some cloud infrastructures even provide support for big data technologies and IoT.

We have tabulated some of the popular cloud services and their support for deep learning, big data technologies and IoT in Table 13.

The expanding possibilities of the cloud have contributed to the growth of Crimeware-as-a-Service (CaaS), which enables cybercriminals with limited technical expertise to conduct organized and automated attacks [170]. There are many types of services provided by CaaS such as, shadow broker services, Neutrino exploit kits, Mirai devices for rent, DiamondFox modular malware services, Tox ransomware-as-a-service, and phishing-as-a-service.

Fig. 6 illustrates the various types of CaaS available.

The above section of this paper had discussed on the state-of-the-art of deep learning, big data technologies and IoT security. Moreover, the support for the three aforementioned domains in the cloud has been discussed. Finally, we introduce on CaaS and some of its types.

6. Open challenges and future directions

This section highlights the most significant research challenges in terms of IoT security using deep learning and big data technologies. The state of-the-art capabilities in IoT security, deep learning, and big data technologies have been examined to determine the major research challenges, suggestions, and future directions.

6.1. Security threat detection

Due to high velocity and variety in multiple domain IoT applications, the complex structure of data makes it more challenging to detect security threats. Further, choosing the recognized set of features for security analytics in deep learning algorithms can be interesting [171]. Existing mechanisms lack efficiency in finding the hidden correlation between these features. Furthermore, emerging deep learning algorithms can handle the hidden parameters from the IoT application.

Table 12
Deep learning, big data technologies and IoT security.

Study	Deep learning	Big data technologies	IoT security
[44]	✓	✓	
[168]	✓	✓	
[78]	✓	✓	
[110]	✓	✓	
[108]	✓	✓	
[81]	✓	✓	
[23]	✓		✓
[24]	✓		✓
[25]	✓		✓
[85]	✓		✓
[80]	✓		✓
[45]	✓		✓
[107]	✓		✓
[82]	✓	✓	
[109]	✓	✓	
[77]	✓	✓	
[89]	✓		✓
[86]	✓		✓
[87]	✓		✓

Table 13
Cloud infrastructure for deep learning, big data technologies and IoT.

Cloud services	Support for deep learning	Support for big data technologies	Support for IoT
Google Cloud	✓	✓	✓
AWS Sagemaker	✓	✓	✓
Deep Cognition	✓	✓	–
IBM Watson	✓	✓	✓
Microsoft Azure	✓	✓	✓
Oracle Cloud	✓	✓	✓
Alibaba Cloud	✓	✓	✓
TensorPad	✓	–	–

Moreover, deep learning is capable of finding the correlation in the variety of data. Additionally, it is possible to acquire high detection rate to detect zero-day attacks more efficiently [172]. Lastly, compared to traditional approaches, the distribution representation of deep learning algorithm can handle multiple feature selection with tremendous data to extract the information for multi-domain IoT applications [173].

6.2. Training duration

Existing techniques take longer time to train the model for accurate detection. As well as, they require large datasets to train the model [174]. These two conditions are major bottlenecks in the current mechanism, however the capability of deep learning algorithms to use less training duration and dataset enables to handle the model efficiently. In addition, batch size might also impact the time consumed for training due to the accumulation of network upon weight update [85]. These challenges should be handled by the option of multiple layering in deep learning, which helps to weigh and recognize the set of specific parameters from datasets. Lastly, the confined processing and storage facilities further hinders the model's training time. In contrast, the big data technologies and cloud based architecture shall enhance the efficacy of the model by curtailing the training duration [175].

6.3. Time complexity

Most of the existing detection techniques have been developed for batch processing application and not for real-time detection. Time complexity plays a vital role in detecting threats in IoT applications, which contains more streaming data. Further, it helps to identify the impact of several attributes involved in security threats. Another study has highlighted that irrespective of using massive real time data the most common existing approaches are ineffective in classifying intrusions as they employ shallow learning [25]. Moreover, these time complexity issues can be resolved easily in deep learning approach by implementing GPU component, as it aids in real time processing and is highly efficient in analysis of the threat in real-time [39]. Furthermore, the employment of Apache Spark or Apache Hadoop is effective in minimizing the time complexity [108].

Fig. 6. CaaS types.

6.4. Computing-in-memory

In-memory processing is a trending development technology for processing the data stored in the in-memory database. It plays a vital

Table 14
Summary of recommendation for research challenges and future research directions.

Challenge	Recommendation	Future research direction	Citation
Time complexity	• Employ optimal features for predicting the total job execution time and for detecting attacks • Control flow paths	• Big data technologies and GPU based architecture • Detection parameter for real-time analysis	[25,39,108,176,177]
Training duration	• Small, medium and large batch size for training • Large volumes of attack data • Extreme and ensemble learning machine-based classification • AE to remove noise	• Adopting deep leaning based LSTM and DBN • Shorter training time • Capable to handle large volume of data for processing and storage	[30,82,85,175,178]
Computing-in-memory	• Streaming analytics • Memory-centric architecture • Data loading	• Real-time processing • Big data technologies • New analytics model	[2,179,180]
Security threat detection	• Distributed or service oriented model to handle high velocity and variety of data • Early detection • Fault tolerance	• Incorporating big data technologies and hybrid deep learning algorithms	[171–173]
Computational and energy constraint	• Low-dimensional domain • Dimensionality reduction • Large scale big data analytics • Distributed algorithms	• Lightweight model • MLLib library in Apache Spark • Edge or cloud computing • Distributed system • High-speed networks (5G)	[25,29,82,181]
Security at edge	• Device identification • Access control • Fault tolerance • Hybrid algorithm for analytics	• Performing data analytics at edge-modernized framework	[45,176,180,182]

role in streaming analytics and memory-centric architecture. Conventional techniques are based on disk storage and relational database which face multiple challenges to handle the modern data volume from IoT devices. Further, these techniques become inadequate to integrate for security analytics which makes the organizations to be more vulnerable in terms of security. In a relational database, data are stored in multiple tables and need to use SQL to do any query processing. These existing approaches further pose difficultly in combining and aggregating the data for processing and SQL is designed to fetch rows of data before processing. The above mentioned issues will be easily handled by in-memory processing for security analytics. The stored data is rapidly accessed when it is saved in RAM or flash memory compared to disk storage. Further, in-memory processing allows data to be analysed in real-time. Real-time processing helps to make faster reporting and decision making for a security threat. Modern big data technologies like Apache Spark and Apache Flink process their data in-memory. Incorporating these technologies to develop new security analytics will enhance the performance and efficiency for security analytics [2,179,180].

6.5. Computational and energy constraint

Computational complexity is one of the most important challenges in the field of IoT device security, deep learning, and big data, research areas. IoT devices are operated in the low power batteries and their CPUs have lower clock rates. Performing any computations in the IoT devices should be fast and shall minimize the straightforward operations [61]. Instead, computation should be carried out in the cloud or edge computing. Similarly, a study has highlighted that implementing algorithmic based security system should focus more on producing lightweight computation system for analysis [182]. On the other hand, the growth of big data as well as increasing computation power benefits the deep learning techniques to grow rapidly, which in turn have been used in serval industries [25]. Further, computations can be optimized using the properties of distributed computing and distributed algorithms. The operations of these algorithm are performed in the hybrid network, in which the jobs are distributed to various machines to improve their efficiency [30]. Some of the above discussed challenges have been easily handled by the Apache Spark streaming big data technological framework, which is capable of utilizing the RDD, Dstreams and parallel computing features to process the data with feasible computation [108].

6.6. Security at edge

Edge computing platform enables more scalability for computational processes and storage power for IoT devices. Further, it provides opportunities to the devices located near to the data sources, which permit intelligent operation to be performed away from the centralized point of infrastructure. Meanwhile, cloud edge infrastructure in the network keeps the IoT data source, especially with regards to network computing to furnish an intelligent edge services to detect any threat in real-time. Moreover, IoT devices do not have sufficient resources to store and analyse the data for any threat [175]. Thus, adopting edge computing will facilitate to handle the above challenges by distributing the process to multiple resource over cloud for analysis [176]. Lastly, integrating deep learning and big data technologies for security analytics of IoT devices provide more efficient processing system to effectively and accurately detect threats.

Table 14 summarizes the challenges, the recommendations, and future research directions.

This section had highlighted on the major research challenges in incorporating deep learning and big data technologies for IoT security. Furthermore, the challenges have been tabulated and recommendation and future research directions have been presented.

7. Conclusion

The expanding population of IoT devices has contributed to the consideration of security risks associated with them. IoT devices are proven to be vulnerable due to the recent increasing attacks such as, the Carna and Mirai botnets. Additionally, IoT devices produce large volume, velocity and variety of data. This makes existing solutions less efficient and require modern day solutions. In this regard, deep learning has been widely accepted amongst researchers and organizations due to their high accuracy, ability to learn deep features, and minimal human supervision. In addition, big data technologies have also been of an interest due to their capability in processing large amounts of data,

along with their capability to process data in a variety of environments such as real-time, batch, and stream. Hence, this study had investigated the possibilities of incorporating deep learning and big data technologies for IoT security. Our findings indicate that many studies have incorporated deep learning with IoT security or deep learning with big data technologies, however, there is a lack of research in incorporating deep learning and big data technologies for IoT security, nevertheless, our investigations had revealed that two studies have proven the efficiency and feasibility of incorporating deep learning and big data technologies for IoT security over traditional models. Considering the various IoT security requirements discussed (see Section 3.3) and the challenges discussed (see Section 6) we have planned to propose a novel framework for IoT security based on deep learning and big data technologies and perform an experimental analysis to prove its efficacy, in the near future. Furthermore, we will attempt to negate the challenges in terms of solving the issues discussed in incorporating deep learning and big data technologies for IoT security.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to extend our sincere gratitude and appreciation to Dr. Ahmed Tajuddin Bin Samsuddin, Mr. Keng Chee Chan, Ir. Dr. Abdul Aziz Bin Abdul Rahman, Mrs. Azlinda Tee Binti Md Azlan Tee, and the members of editorial board, Telekom Research & Development Sdn. Bhd for their continuous support towards the publication of this manuscript. We would also like to thank the anonymous reviewers and the editors of this journal who helped us in improving the quality of our manuscript.

References

[1] N. Mohan, J. Kangasharju, Edge-fog cloud: A distributed cloud for internet of things computations, in: Proc. Cloudification of the Internet of Things (CIoT), 2016, pp. 1–6, http://dx.doi.org/10.1109/CIOT.2016.7872914.

[2] R.A.A. Habeeb, F. Nasaruddin, A. Gani, I.A.T. Hashem, E. Ahmed, M. Imran, Real-time big data processing for anomaly detection: A survey, Int. J. Inf. Manage. 45 (2019) 289–307.

[3] G. Davis, G. Davis, Trending: Iot malware attacks of 2018, 2018, accessed on 10 May 2019. URL https://securingtomorrow.mcafee.com/consumer/mobile-and-iot-security/top-trending-iot-malware-attacks-of-2018/.

[4] W.G. Wong, Developers discuss iot security and platforms trends, 2015, accessed on 1 May 2019. URL https://www.electronicdesign.com/embedded/developers-discuss-iot-security-and-platforms-trends.

[5] New trends in the world of iot threats, 2019, accessed on 10 May 2019. URL https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/.

[6] A. Katal, M. Wazid, R.H. Goudar, Big data: Issues, challenges, tools and good practices, in: 2013 Sixth International Conference on Contemporary Computing (IC3), IEEE, 2013, http://dx.doi.org/10.1109/ic3.2013.6612229.

[7] A.A. Cardenas, P.K. Manadhata, S.P. Rajan, Big data analytics for security, IEEE Secur. Priv. 11 (6) (2013) 74–76, http://dx.doi.org/10.1109/msp.2013.138.

[8] C.D. McDermott, F. Majdani, A.V. Petrovski, Botnet detection in the internet of things using deep learning approaches, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[9] M. Aly, F. Khomh, M. Haoues, A. Quintero, S. Yacout, Enforcing security in internet of things frameworks: A systematic literature review, Internet Things (2019) 100050.

[10] J. Pan, Z. Yang, Cybersecurity challenges and opportunities in the new edge computing+ iot world, in: Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, ACM, 2018, pp. 29–32.

[11] A.F.A. Rahman, M. Daud, M.Z. Mohamad, Securing sensor to cloud ecosystem using internet of things (iot) security framework, in: Proceedings of the International Conference on Internet of Things and Cloud Computing, ACM, 2016, p. 79.

[12] C. Perera, R. Ranjan, L. Wang, S.U. Khan, A.Y. Zomaya, Big data privacy in the internet of things era, IT Prof. 17 (3) (2015) 32–39.

[13] R. Hussain, I. Abdullah, Review of different encryptionand decryption techniques used for security and privacy of iot in different applications, in: 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), IEEE, 2018, pp. 293–297.

[14] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for iot big data and streaming analytics: A survey, IEEE Commun. Surv. Tutor. 20 (4) (2018) 2923–2960.

[15] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I.A.T. Hashem, A. Siddiqa, I. Yaqoob, Big iot data analytics: architecture, opportunities, and open research challenges, IEEE Access 5 (2017) 5247–5261.

[16] A. Alnasser, H. Sun, J. Jiang, Cyber security challenges and solutions for v2x communications: A survey, Comput. Netw. 151 (2019) 52–67.

[17] P.I.R. Grammatikis, P.G. Sarigiannidis, I.D. Moscholios, Securing the internet of things: Challenges, threats and solutions, Internet Things 5 (2019) 41–70.

[18] J. Deogirikar, A. Vidhate, Security attacks in iot: A survey, in: Proc. Analytics and Cloud (I-SMAC) 2017 Int. Conf. I-SMAC (IoT in Social, Mobile, 2017, pp. 32–37, http://dx.doi.org/10.1109/I-SMAC.2017.8058363.

[19] A.O. Otuoze, M.W. Mustafa, R.M. Larik, Smart grids security challenges: Classification by sources of threats, J. Electr. Syst. Inf. Technol. 5 (3) (2018) 468–483.

[20] B. Kolosnjaji, A. Zarras, G. Webster, C. Eckert, Deep learning for classification of malware system call sequences, in: Australasian Joint Conference on Artificial Intelligence, Springer, 2016, pp. 137–149.

[21] Z. Yuan, Y. Lu, Z. Wang, Y. Xue, Droid-sec: deep learning in android malware detection, in: ACM SIGCOMM Computer Communication Review, Vol. 44, ACM, 2014, pp. 371–372.

[22] Z. Yuan, Y. Lu, Y. Xue, Droiddetector: android malware characterization and detection using deep learning, Tsinghua Sci. Technol. 21 (1) (2016) 114–123, http://dx.doi.org/10.1109/TST.2016.7399288.

[23] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, N-baiot—network-based detection of iot botnet attacks using deep autoencoders, IEEE Pervasive Comput. 17 (3) (2018) 12–22.

[24] A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for internet of things, Future Gener. Comput. Syst. 82 (2018) 761–768.

[25] B. Roy, H. Cheung, A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network, in: 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), IEEE, 2018, pp. 1–6.

[26] R.A. Ariyaluran Habeeb, F. Nasaruddin, A. Gani, M.A. Amanullah, I. Abaker Targio Hashem, E. Ahmed, M. Imran, Clustering-based real-time anomaly detection—A breakthrough in big data technologies, Trans. Emerg. Telecommun. Technol., 0 (0) e3647, e3647 ett.3647, http://dx.doi.org/10.1002/ett.3647.

[27] G.P. Gupta, M. Kulariya, A framework for fast and efficient cyber security network intrusion detection using apache spark, Procedia Comput. Sci. 93 (2016) 824–831.

[28] V.P. Janeja, A. Azari, J.M. Namayanja, B. Heilig, B-dids: Mining anomalies in a big-distributed intrusion detection system, in: 2014 IEEE International Conference on Big Data (Big Data), IEEE, 2014, pp. 32–34.

[29] K. Vimalkumar, N. Radhika, A big data framework for intrusion detection in smart grids using apache spark, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2017, pp. 198–204.

[30] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, S. Venkatraman, Deep learning approach for intelligent intrusion detection system, IEEE Access 7 (2019) 41525–41550.

[31] C. Cimpanu, Sirenjack attack lets hackers take control over emergency alert sirens, 2018, accessed on 10 May 2019. URL https://www.bleepingcomputer.com/news/security/sirenjack-attack-lets-hackers-take-control-over-emergency-alert-sirens/.

[32] J. Sanders, 5 biggest iot security failures of 2018, 2019, accessed on 1 May 2019. URL https://www.techrepublic.com/article/5-biggest-iot-security-failures-of-2018/.

[33] L. Mathews, Hackers use ddos attack to cut heat to apartments, 2016, accessed on 1 May 2019. URL https://www.forbes.com/sites/leemathews/2016/11/07/ddos-attack-leaves-finnish-apartments-without-heat#4bd0d0961a09.

[34] Iot role in dyn cyberattack, 2019, accessed on 10 May 2019. URL https://www.kaspersky.com/blog/attack-on-dyn-explained/13325/.

[35] D. Etherington, K. Conger, D. Etherington, K. Conger, Large ddos attacks cause outages at twitter, spotify, and other sites – techcrunch, 2016, accessed on 10 May 2019. URL https://techcrunch.com/2016/10/21/many-sites-including-twitter-and-spotify-suffering-outage/.

[36] The possible vendetta behind the east coast web slowdown, 2019, accessed on 10 May 2019. URL https://www.bloomberg.com/news/articles/2016-10-21/internet-service-disrupted-in-large-parts-of-eastern-u-s.

[37] A. Schiffer, How a fish tank helped hack a casino, 2017, accessed on 1 May 2019. URL https://www.washingtonpost.com/news/innovations/wp/2017/07/21/how-a-fish-tank-helped-hack-a-casino/?utm_term=.8ba4c46540ef.

[38] T. Dodrill, Hacker turns baby monitor into real life nightmare, 2014, accessed on 1 May 2019. URL https://www.offthegridnews.com/privacy/hacker-turns-baby-monitor-into-real-life-nightmare/.
[39] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48.
[40] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2016, pp. 372–387.
[41] R. Shokri, V. Shmatikov, Privacy-preserving deep learning, in: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15, ACM, New York, NY, USA, 2015, pp. 1310–1321, http://dx.doi.org/10.1145/2810103.2813687.
[42] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, Deep learning for sensor-based activity recognition: A survey, Pattern Recognit. Lett. 119 (2019) 3–11.
[43] M. Strohbach, H. Ziekow, V. Gazis, N. Akiva, Towards a big data analytics framework for iot and smart city applications, in: Modeling and Processing for Next-Generation Big-Data Technologies, Springer, 2015, pp. 257–282.
[44] G. Mylavarapu, J. Thomas, A.K. TK, Real-time hybrid intrusion detection system using apache storm, in: Proc. and 2015 IEEE 12th Int 2015 IEEE 17th Int. Conf. High Performance Computing and Communications IEEE 7th Int. Symp. Cyberspace Safety and Security Conf. Embedded Software and Systems, 2015, pp. 1436–1441, http://dx.doi.org/10.1109/HPCC-CSS-ICESS.2015.241.
[45] Y. Zhou, M. Han, L. Liu, J.S. He, Y. Wang, Deep learning approach for cyberattack detection, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2018, pp. 262–267.
[46] C. Wang, S. Dong, X. Zhao, G. Papanastasiou, H. Zhang, G. Yang, Saliencygan: Deep learning semi-supervised salient object detection in the fog of iot, IEEE Trans. Ind. Inf. (2019).
[47] A. Gandomi, M. Haider, Beyond the hype: Big data concepts, methods, and analytics, Int. J. Inf. Manag. 35 (2) (2015) 137–144.
[48] K. Adam, M.A.I. Fakharaldien, J.M. Zain, M.A. Majid, A. Noraziah, Bigdata: Issues, challenges, technologies and methods, in: Proceedings of the International Conference on Data Engineering 2015 (DaEng-2015), Springer, 2019, pp. 541–550.
[49] V.K. Vavilapalli, A.C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, et al., Apache hadoop yarn: Yet another resource negotiator, in: Proceedings of the 4th Annual Symposium on Cloud Computing, ACM, 2013, p. 5.
[50] M. Zaharia, R.S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M.J. Franklin, et al., Apache spark: a unified engine for big data processing, Commun. ACM 59 (11) (2016) 56–65.
[51] J.S. van der Veen, B. van der Waaij, E. Lazovik, W. Wijbrandi, R.J. Meijer, Dynamically scaling apache storm for the analysis of streaming data, in: 2015 IEEE First International Conference on Big Data Computing Service and Applications, IEEE, 2015, pp. 154–161.
[52] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, K. Tzoumas, Apache flink: Stream and batch processing in a single engine, Bull. IEEE Comput. Soc. Tech. Comm. Data Eng. 36 (4) (2015).
[53] A. Chebotko, A. Kashlev, S. Lu, A big data modeling methodology for apache cassandra, in: 2015 IEEE International Congress on Big Data, IEEE, 2015, pp. 238–245.
[54] D. Borthakur, J. Gray, J.S. Sarma, K. Muthukkaruppan, N. Spiegelberg, H. Kuang, K. Ranganathan, D. Molkov, A. Menon, S. Rash, et al., Apache hadoop goes realtime at facebook, in: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, ACM, 2011, pp. 1071–1080.
[55] S. Moin, A. Karim, Z. Safdar, K. Safdar, E. Ahmed, M. Imran, Securing iots in distributed blockchain: Analysis, requirements and open issues, Future Gener. Comput. Syst. 100 (2019) 325–343.
[56] F.X. Ming, R.A.A. Habeeb, F.H.B. Md Nasaruddin, A.B. Gani, Real-time carbon dioxide monitoring based on iot & cloud technologies, in: Proceedings of the 2019 8th International Conference on Software and Computer Applications, ACM, 2019, pp. 517–521.
[57] M.A. Khan, K. Salah, Iot security: Review, blockchain solutions, and open challenges, Future Gener. Comput. Syst. 82 (2018) 395–411.
[58] J.-S. Cho, S.-S. Yeo, S.K. Kim, Securing against brute-force attack: A hash-based rfid mutual authentication protocol using a secret value, Comput. Commun. 34 (3) (2011) 391–397.
[59] H.A. Khattak, M.A. Shah, S. Khan, I. Ali, M. Imran, Perception layer security in internet of things, Future Gener. Comput. Syst. 100 (2019) 144–164.
[60] Spark security, 2019, accessed on 10 May 2019. URL https://spark.apache.org/docs/latest/security.html.
[61] M.M. Hossain, M. Fotouhi, R. Hasan, Towards an analysis of security issues, challenges, and open problems in the internet of things, in: 2015 IEEE World Congress on Services, IEEE, 2015, pp. 21–28.
[62] How-to: Do data quality checks using apache spark dataframes, 2015, accessed on 10 May 2019. URL https://blog.cloudera.com/blog/2015/07/how-to-do-data-quality-checks-using-apache-spark-dataframes/.
[63] S. Babar, P. Mahalle, A. Stango, N. Prasad, R. Prasad, Proposed security model and threat taxonomy for the internet of things (iot), in: International Conference on Network Security and Applications, Springer, 2010, pp. 420–429.
[64] Spark standalone mode, 2019, accessed on 10 May 2019. URL https://spark.apache.org/docs/latest/spark-standalone.html#high-availability.
[65] G. Huang, S. Song, J.N. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. 44 (12) (2014) 2405–2417.
[66] C.-Y. Liou, J.-C. Huang, W.-C. Yang, Modeling word perception using the elman network, Neurocomputing 71 (16–18) (2008) 3150–3157.
[67] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, D.-R. Liou, Autoencoder for words, Neurocomputing 139 (2014) 84–96, http://dx.doi.org/10.1016/j.neucom.2013.09.055.
[68] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, M. Ghogho, Deep recurrent neural network for intrusion detection in sdn-based networks, in: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), IEEE, 2018, pp. 202–206.
[69] B. Li, M.H. Najafi, D.J. Lilja, Using stochastic computing to reduce the hardware requirements for a restricted boltzmann machine classifier, in: Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, ACM, 2016, pp. 36–41.
[70] A.M. Abdel-Zaher, A.M. Eldeib, Breast cancer classification using deep belief networks, Expert Syst. Appl. 46 (2016) 139–144.
[71] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554, http://dx.doi.org/10.1162/neco.2006.18.7.1527.
[72] G. Hinton, Deep belief networks, Scholarpedia 4 (5) (2009) 5947, http://dx.doi.org/10.4249/scholarpedia.5947.
[73] H. Sak, A. Senior, F. Beaufays, Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in: Fifteenth Annual Conference of the International Speech Communication Association, 2014.
[74] Y. LeCun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W.E. Hubbard, L.D. Jackel, Handwritten digit recognition with a back-propagation network, in: Advances in Neural Information Processing Systems, 1990, pp. 396–404.
[75] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551, http://dx.doi.org/10.1162/neco.1989.1.4.541.
[76] M. Liang, X. Hu, Recurrent convolutional neural network for object recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3367–3375.
[77] S. Alzahrani, L. Hong, Detection of distributed denial of service (ddos) attacks using artificial intelligence on cloud, in: 2018 IEEE World Congress on Services (SERVICES), IEEE, 2018, pp. 35–36.
[78] A. Thilina, S. Attanayake, S. Samarakoon, D. Nawodya, L. Rupasinghe, N. Pathirage, T. Edirisinghe, K. Krishnadeva, Intruder detection using deep learning and association rule mining, in: 2016 IEEE International Conference on Computer and Information Technology (CIT), IEEE, 2016, pp. 615–620.
[79] A. Elsaeidy, I. Elgendi, K.S. Munasinghe, D. Sharma, A. Jamalipour, A smart city cyber security platform for narrowband networks, in: Proc. 27th Int. Telecommunication Networks and Applications Conf. (ITNAC), 2017, pp. 1–6, http://dx.doi.org/10.1109/ATNAC.2017.8215388.
[80] A. Dawoud, S. Shahristani, C. Raun, Deep learning and software-defined networks: Towards secure iot architecture, Internet Things 3 (2018) 82–89.
[81] L. Dong-Lan, L. Xin, Y. Hao, W. Wen-Ting, Z. Xiao-Hong, C. Jian-Fei, A multi-level deep learning method for data fusion and anomaly detection of power big data, in: 3rd Annual International Conference on Electronics, Electrical Engineering and Information Science (EEEIS 2017), Atlantis Press, 2017.
[82] N. Marir, H. Wang, G. Feng, B. Li, M. Jia, Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark, IEEE Access 6 (2018) 59657–59671.
[83] Y. He, G.J. Mendis, J. Wei, Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism, IEEE Trans. Smart Grid 8 (5) (2017) 2505–2516, http://dx.doi.org/10.1109/tsg.2017.2703842.
[84] J. Chauhan, S. Seneviratne, Y. Hu, A. Misra, A. Seneviratne, Y. Lee, Breathing-based authentication on resource-constrained IoT devices using recurrent neural networks, Computer 51 (5) (2018) 60–67, http://dx.doi.org/10.1109/mc.2018.2381119.
[85] H. HaddadPajouh, A. Dehghantanha, R. Khayami, K.-K.R. Choo, A deep recurrent neural network based approach for internet of things malware threat hunting, Future Gener. Comput. Syst. 85 (2018) 88–96, http://dx.doi.org/10.1016/j.future.2018.03.007.
[86] M. Roopak, G.Y. Tian, J. Chambers, Deep learning models for cyber security in iot networks, in: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2019, pp. 0452–0457.
[87] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, R. Khayami, K.-K.R. Choo, D.E. Newton, DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer, Future Gener. Comput. Syst. 90 (2019) 94–104, http://dx.doi.org/10.1016/j.future.2018.07.045.
[88] J. Su, V.D. Vasconcellos, S. Prasad, S. Daniele, Y. Feng, K. Sakurai, Lightweight classification of IoT malware based on image recognition, in: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), IEEE, 2018, http://dx.doi.org/10.1109/compsac.2018.10315.

[89] A. Azmoodeh, A. Dehghantanha, K.R. Choo, Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning, IEEE Trans. Sustain. Comput. 4 (1) (2019) 88–95, http://dx.doi.org/10.1109/TSUSC.2018.2809665.
[90] W.G. Hatcher, W. Yu, A survey of deep learning: Platforms, applications and emerging research trends, IEEE Access 6 (2018) 24411–24432, http://dx.doi.org/10.1109/ACCESS.2018.2830661.
[91] H. Ma, F. Mao, G.W. Taylor, Theano-mpi: a theano-based distributed training framework, in: European Conference on Parallel Processing, Springer, 2016, pp. 800–813.
[92] P. Roy, S.L. Song, S. Krishnamoorthy, A. Vishnu, D. Sengupta, X. Liu, NUMA-Caffe: NUMA-aware deep learning neural networks, ACM Trans. Archit. Code Optim. (TACO) 15 (2) (2018) 24.
[93] N. Ketkar, Introduction to pytorch, in: Deep Learning with Python, Springer, 2017, pp. 195–208.
[94] M. Ravanelli, T. Parcollet, Y. Bengio, The pytorch-kaldi speech recognition toolkit, in: Proc. Speech and Signal Processing (ICASSP) ICASSP 2019 - 2019 IEEE Int. Conf. Acoustics, 2019, pp. 6465–6469, http://dx.doi.org/10.1109/ICASSP.2019.8683713.
[95] F. Seide, A. Agarwal, CNTK: Microsoft's open-source deep-learning toolkit, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '16, ACM, New York, NY, USA, 2016, p. 2135, http://dx.doi.org/10.1145/2939672.2945397.
[96] A. Candel, V. Parmar, E. LeDell, A. Arora, Deep Learning with H2O, H2O. ai Inc., 2016.
[97] A. Parvat, J. Chavan, S. Kadam, S. Dev, V. Pathak, A survey of deep-learning frameworks, in: Proc. Int. Conf. Inventive Systems and Control (ICISC), 2017, pp. 1–7, http://dx.doi.org/10.1109/ICISC.2017.8068684.
[98] What is a confusion matrix in machine learning, 2018, accessed on 10 May 2019. URL https://machinelearningmastery.com/confusion-matrix-machine-learning/.
[99] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, third ed., Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.
[100] D.M. Powers, Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation, J. Mach. Learn. Technol. 2 (1) (2011) 37–63.
[101] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manage. 45 (4) (2009) 427–437, http://dx.doi.org/10.1016/j.ipm.2009.03.002.
[102] What is a false positive rate?, 2019, accessed on 10 May 2019. URL https://www.corvil.com/kb/what-is-a-false-positive-rate.
[103] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, C. Wang, Machine learning and deep learning methods for cybersecurity, IEEE Access 6 (2018) 35365–35381, http://dx.doi.org/10.1109/ACCESS.2018.2836950.
[104] Matthews correlation coefficient, 2019, accessed on 10 May 2019. URL https://scikit-learn.org/stable/modules/generated/sklearn.metrics.matthews_corrcoef.html.
[105] B. Matthews, Comparison of the predicted and observed secondary structure of t4 phage lysozyme, Biochim. Biophys. Acta (BBA) - Protein Struct. 405 (2) (1975) 442–451, http://dx.doi.org/10.1016/0005-2795(75)90109-9.
[106] M.L. McHugh, Interrater reliability: the kappa statistic, Biochem. Medica: Biochem. Medica 22 (3) (2012) 276–282.
[107] O. Brun, Y. Yin, E. Gelenbe, Deep learning with dense random neural network for detecting attacks against iot-connected home environments, Procedia Comput. Sci. 134 (2018) 458–463, http://dx.doi.org/10.1016/j.procs.2018.07.183.
[108] A. Gupta, H.K. Thakur, R. Shrivastava, P. Kumar, S. Nag, A big data analysis framework using apache spark and deep learning, in: Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW), 2017, pp. 9–16, http://dx.doi.org/10.1109/ICDMW.2017.9.
[109] R. Kozik, Distributing extreme learning machines with apache spark for netflow-based malware activity detection, Pattern Recognit. Lett. 101 (2018) 14–20.
[110] C.-J. Hsieh, T.-Y. Chan, Detection ddos attacks based on neural-network using apache spark, in: 2016 International Conference on Applied System Innovation (ICASI), IEEE, 2016, pp. 1–4.
[111] S. Rathore, J.H. Park, Semi-supervised learning based distributed attack detection framework for iot, Appl. Soft Comput. 72 (2018) 79–89.
[112] R. Abdulhammed, M. Faezipour, A. Abuzneid, A. AbuMallouh, Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic, IEEE Sensors Lett. 3 (1) (2019) 1–4, http://dx.doi.org/10.1109/lsens.2018.2879990.
[113] M. Ahmed, A.N. Mahmood, J. Hu, A survey of network anomaly detection techniques, J. Netw. Comput. Appl. 60 (2016) 19–31.
[114] M. Nobakht, V. Sivaraman, R. Boreli, A host-based intrusion detection and mitigation framework for smart home iot using openflow, in: 2016 11th International Conference on Availability, Reliability and Security (ARES), IEEE, 2016, pp. 147–156.
[115] J. Saxe, K. Berlin, Deep neural network based malware detection using two dimensional binary program features, in: 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), IEEE, 2015, pp. 11–20.
[116] I. Kara, M. Aydos, Static and dynamic analysis of third generation cerber ransomware, in: 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), IEEE, 2018, pp. 12–17.
[117] J.M. Ceron, K. Steding-Jessen, C. Hoepers, L.Z. Granville, C.B. Margi, Improving iot botnet investigation using an adaptive network layer, Sensors 19 (3) (2019) 727.
[118] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, Ddos in the iot: Mirai and other botnets, Computer 50 (7) (2017) 80–84.
[119] S. Vashi, J. Ram, J. Modi, S. Verma, C. Prakash, Internet of things (iot): A vision, architectural elements, and security issues, in: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), IEEE, 2017, pp. 492–496.
[120] I. Andrea, C. Chrysostomou, G. Hadjichristofi, Internet of things: Security vulnerabilities and challenges, in: 2015 IEEE Symposium on Computers and Communication (ISCC), IEEE, 2015, pp. 180–187.
[121] B. Barak, Constant-round coin-tossing with a man in the middle or realizing the shared random string model, in: The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings, IEEE, 2002, pp. 345–355.
[122] V. Ramachandran, S. Nandi, Detecting ARP spoofing: An active technique, in: International Conference on Information Systems Security, Springer, 2005, pp. 239–250.
[123] S. Son, V. Shmatikov, The hitchhiker's guide to dns cache poisoning, in: International Conference on Security and Privacy in Communication Systems, Springer, 2010, pp. 466–483.
[124] P. De Ryck, L. Desmet, F. Piessens, W. Joosen, Secsess: Keeping your session tucked away in your browser, in: Proceedings of the 30th Annual ACM Symposium on Applied Computing, ACM, 2015, pp. 2171–2176.
[125] K. Sonar, H. Upadhyay, A survey: Ddos attack on internet of things, Int. J. Eng. Res. Dev. 10 (11) (2014) 58–63.
[126] A. Bijalwan, M. Wazid, E.S. Pilli, R.C. Joshi, Forensics of random-udp flooding attacks, J. Netw. 10 (5) (2015) 287.
[127] M. Beaumont-Gay, A comparison of syn flood detection algorithms, in: Second International Conference on Internet Monitoring and Protection (ICIMP 2007), IEEE, 2007, p. 9.
[128] J. Erickson, Hacking: the art of exploitation, No starch press, 2008.
[129] Y.G. Dantas, V. Nigam, I.E. Fonseca, A selective defense for application layer ddos attacks, in: 2014 IEEE Joint Intelligence and Security Informatics Conference, IEEE, 2014, pp. 75–82.
[130] J. Krupp, M. Backes, C. Rossow, Identifying the scan and attack infrastructures behind amplification ddos attacks, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 1426–1437.
[131] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, A. Jamalipour, A survey of routing attacks in mobile ad hoc networks, IEEE Wirel. Commun. 14 (5) (2007) 85–91.
[132] Z. Trifa, M. Khemakhem, Sybil nodes as a mitigation strategy against sybil attack, Procedia Comput. Sci. 32 (2014) 1135–1140.
[133] I. Krontiris, T. Dimitriou, T. Giannetsos, M. Mpasoukos, Intrusion detection of sinkhole attacks in wireless sensor networks, in: International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics, Springer, 2007, pp. 150–161.
[134] L. Wallgren, S. Raza, T. Voigt, Routing attacks and countermeasures in the rpl-based internet of things, Int. J. Distrib. Sens. Netw. 9 (8) (2013) 794326.
[135] C. Eik Loo, M. Yong Ng, C. Leckie, M. Palaniswami, Intrusion detection for routing attacks in sensor networks, Int. J. Distrib. Sens. Netw. 2 (4) (2006) 313–332.
[136] P. Pongle, G. Chavan, Real time intrusion and wormhole attack detection in internet of things, Int. J. Comput. Appl. 121 (9) (2015).
[137] M. Mahajan, K. Reddy, M. Rajput, Design and simulation of a blacklisting technique for detection of hello flood attack on leach protocol, Procedia Comput. Sci. 79 (2016) 675–682.
[138] N. Gruschka, M. Jensen, Attack surfaces: A taxonomy for attacks on cloud services, in: 2010 IEEE 3rd International Conference on Cloud Computing, IEEE, 2010, pp. 276–279.
[139] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, A survey of intrusion detection techniques in cloud, J. Netw. Comput. Appl. 36 (1) (2013) 42–57.
[140] J. Liu, Y. Xiao, C.P. Chen, Authentication and access control in the internet of things, in: 2012 32nd International Conference on Distributed Computing Systems Workshops, IEEE, 2012, pp. 588–592.
[141] C. Herley, D. Florêncio, Protecting financial institutions from brute-force attacks, in: IFIP International Information Security Conference, Springer, 2008, pp. 681–685.
[142] S. Chakrabarti, M. Singhal, Password-based authentication: Preventing dictionary attacks, Computer 40 (6) (2007) 68–74.
[143] D.F. Smith, A. Wiliem, B.C. Lovell, Face recognition on consumer devices: Reflections on replay attacks, IEEE Trans. Inf. Forensics Secur. 10 (4) (2015) 736–745.
[144] S. Gajek, M. Jensen, L. Liao, J. Schwenk, Analysis of signature wrapping attacks and countermeasures, in: 2009 IEEE International Conference on Web Services, IEEE, 2009, pp. 575–582.

[145] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, R. Atkinson, Threat analysis of iot networks using artificial neural network intrusion detection system, in: 2016 International Symposium on Networks, Computers and Communications (ISNCC), IEEE, 2016, pp. 1–6.
[146] P. Visu, L. Lakshmanan, V. Murugananthan, M.V. Cruz, Software-defined forensic framework for malware disaster management in internet of thing devices for extreme surveillance, Comput. Commun. 147 (2019) 14–20.
[147] M. Jakobsson, S. Myers, Phishing and countermeasures: understanding the increasing problem of electronic identity theft, John Wiley & Sons, 2006.
[148] J.A. Chaudhry, S.A. Chaudhry, R.G. Rittenhouse, Phishing attacks and defenses, Int. J. Secur. Appl. 10 (1) (2016) 247–256.
[149] G.S. Kc, A.D. Keromytis, V. Prevelakis, Countering code-injection attacks with instruction-set randomization, in: Proceedings of the 10th ACM Conference on Computer and Communications Security, ACM, 2003, pp. 272–280.
[150] A. Kieyzun, P.J. Guo, K. Jayaraman, M.D. Ernst, Automatic creation of sql injection and cross-site scripting attacks, in: Proceedings of the 31st International Conference on Software Engineering, IEEE Computer Society, 2009, pp. 199–209.
[151] T. Jim, N. Swamy, M. Hicks, Defeating script injection attacks with browser-enforced embedded policies, in: Proceedings of the 16th International Conference on World Wide Web, ACM, 2007, pp. 601–610.
[152] W. Gao, T. Morris, B. Reaves, D. Richey, On scada control system command and response injection and intrusion detection, in: 2010 ECrime Researchers Summit, IEEE, 2010, pp. 1–9.
[153] J. Horchert, J. Horchert, Mapping the internet: A hacker's secret internet census - spiegel online - international, 2013, accessed on 10 May 2019. URL https://www.spiegel.de/international/world/hacker-measures-the-internet-illegally-with-carna-botnet-a-890413.html.
[154] A. Kleinman, A. Kleinman, The most detailed map of the internet was made by breaking the law, 2017, accessed on 10 May 2019. URL https://www.huffpost.com/entry/internet-map_n_2926934.
[155] Sea turtle DNS hijacking and more weekly news, 2019, accessed on 10 May 2019. URL https://blog.avast.com/sea-turtle-dns-hijacking.
[156] V.A. Almeida, D. Doneda, J. de Souza Abreu, Cyberwarfare and digital governance, IEEE Internet Comput. 21 (2) (2017) 68–71.
[157] N. Moustafa, J. Slay, Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set), in: Proc. Military Communications and Information Systems Conf. (MilCIS), 2015, pp. 1–6, http://dx.doi.org/10.1109/MilCIS.2015.7348942.
[158] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the kdd cup 99 data set, in: Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications, 2009, pp. 1–6, http://dx.doi.org/10.1109/CISDA.2009.5356528.
[159] S.J. Stolfo, Wei Fan, Wenke Lee, A. Prodromidis, P.K. Chan, Cost-based modeling for fraud and intrusion detection: results from the jam project, in: Proc. DARPA Information Survivability Conf. and Exposition. DISCEX'00, Vol. 2, 2000, pp. 130–144 vol.2, http://dx.doi.org/10.1109/DISCEX.2000.821515.
[160] R.P. Lippmann, D.J. Fried, I. Graf, J.W. Haines, K.R. Kendall, D. McClung, D. Weber, S.E. Webster, D. Wyschogrod, R.K. Cunningham, M.A. Zissman, Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation, in: Proc. DARPA Information Survivability Conf. and Exposition. DISCEX'00, Vol. 2, 2000, pp. 12–26 vol.2, http://dx.doi.org/10.1109/DISCEX.2000.821506.
[161] I. Almomani, B. Al-Kasasbeh, M. AL-Akhras, WSN-DS: A dataset for intrusion detection systems in wireless sensor networks, J. Sens. 2016 (2016) 1–16, http://dx.doi.org/10.1155/2016/4731953.
[162] Y.M.P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, C. Rossow, IoTPOT: A novel honeypot for revealing current IoT threats, J. Inf. Process. 24 (3) (2016) 522–533, http://dx.doi.org/10.2197/ipsjjip.24.522.
[163] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, K. Nakao, Statistical analysis of honeypot data and building of kyoto 2006+ dataset for nids evaluation, in: Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, in: BADGERS '11, ACM, New York, NY, USA, 2011, pp. 29–36, http://dx.doi.org/10.1145/1978672.1978676.
[164] D. Aksu, S. Üstebay, M.A. Aydin, T. Atmaca, Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm, Comput. Inf. Sci. (2018).
[165] A. Verma, V. Ranga, Statistical analysis of CIDDS-001 dataset for network intrusion detection systems using distance-based machine learning, Procedia Comput. Sci. 125 (2018) 709–716, http://dx.doi.org/10.1016/j.procs.2017.12.091.
[166] B.A. Tama, K.-H. Rhee, Attack classification analysis of iot network via deep learning approach, Res. Briefs Inf. Commun. Technol. Evol. 3 (15) (2017) 1–9.
[167] I. Mavridis, H. Karatza, Performance evaluation of cloud-based log file analysis with apache hadoop and apache spark, J. Syst. Softw. 125 (2017) 133–151.
[168] A. Khumoyun, Y. Cui, L. Hanku, Spark based distributed deep learning framework for big data applications, in: 2016 International Conference on Information Science and Communications Technologies (ICISCT), IEEE, 2016, pp. 1–5.
[169] S. Kak, New algorithms for training feedforward neural networks, Pattern Recognit. Lett. 15 (3) (1994) 295–298.
[170] A.K. Sood, R.J. Enbody, Crimeware-as-a-service—a survey of commoditized crimeware in the underground market, Int. J. Crit. Infrastruct. Prot. 6 (1) (2013) 28–38.
[171] D. Ravi, C. Wong, B. Lo, G.-Z. Yang, A deep learning approach to on-node sensor data analytics for mobile or wearable devices, IEEE J. Biomed. Health Inf. 21 (1) (2016) 56–64.
[172] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, M. Ghogho, Deep learning approach for network intrusion detection in software defined networking, in: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), IEEE, 2016, pp. 258–263.
[173] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, M. Zhu, Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection, IEEE Access 6 (2017) 1792–1806.
[174] Z. Zhao, A. Kumar, Accurate periocular recognition under less constrained environment using semantics-assisted convolutional neural network, IEEE Trans. Inf. Forensics Secur. 12 (5) (2016) 1017–1030.
[175] R. Kozik, M. Choraś, M. Ficco, F. Palmieri, A scalable distributed machine learning approach for attack detection in edge computing environments, J. Parallel Distrib. Comput. (ISSN: 0743-7315) 119 (2018) 18–26.
[176] Z. Lu, N. Wang, J. Wu, M. Qiu, IoTDeM: An IoT big data-oriented MapReduce performance prediction extended model in multiple edge clouds, J. Parallel Distrib. Comput. 118 (2018) 316–327, http://dx.doi.org/10.1016/j.jpdc.2017.11.001.
[177] L. Bao, Q. Li, P. Lu, J. Lu, T. Ruan, K. Zhang, Execution anomaly detection in large-scale systems through console log analysis, J. Syst. Softw. 143 (2018) 172–186, http://dx.doi.org/10.1016/j.jss.2018.05.016.
[178] D. Berman, A. Buczak, J. Chavis, C. Corbett, A survey of deep learning methods for cyber security, Information 10 (4) (2019) 122, http://dx.doi.org/10.3390/info10040122.
[179] I. Kotenko, I. Saenko, A. Kushnerevich, A. Branitskiy, Attack detection in IoT critical infrastructures: A machine learning and big data processing approach, in: 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), IEEE, 2019, http://dx.doi.org/10.1109/empdp.2019.8671571.
[180] D.C. Mocanu, E. Mocanu, P.H. Nguyen, M. Gibescu, A. Liotta, Big IoT data mining for real-time energy disaggregation in buildings, in: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2016, http://dx.doi.org/10.1109/smc.2016.7844820.
[181] W. Wang, M. Zhu, J. Wang, X. Zeng, Z. Yang, End-to-end encrypted traffic classification with one-dimensional convolution neural networks, in: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), IEEE, 2017, http://dx.doi.org/10.1109/isi.2017.8004872.
[182] I. Kotenko, I. Saenko, A. Branitskiy, Framework for mobile internet of things security monitoring based on big data processing and machine learning, IEEE Access 6 (2018) 72714–72723, http://dx.doi.org/10.1109/access.2018.2881998.