A survey on application of machine learning for Internet of Things

Article *in* International Journal of Machine Learning and Cybernetics · August 2018

DOI: 10.1007/s13042-018-0834-5

Laizhong Cui
Shenzhen University
88 PUBLICATIONS **1,160** CITATIONS

Zhong Ming
Shenzhen University
195 PUBLICATIONS **2,864** CITATIONS

International Journal of Machine Learning and Cybernetics (2018) 9:1399–1417
https://doi.org/10.1007/s13042-018-0834-5

ORIGINAL ARTICLE

A survey on application of machine learning for Internet of Things

Laizhong Cui[1] · Shu Yang[1] · Fei Chen[1] · Zhong Ming[1] · Nan Lu[1] · Jing Qin[2]

Received: 17 April 2018 / Accepted: 29 May 2018 / Published online: 11 June 2018

Abstract

Internet of Things (IoT) has become an important network paradigm and there are lots of smart devices connected by IoT. IoT systems are producing massive data and thus more and more IoT applications and services are emerging. Machine learning, as an another important area, has obtained a great success in several research fields such as computer vision, computer graphics, natural language processing, speech recognition, decision-making, and intelligent control. It has also been introduced in networking research. Many researches study how to utilize machine learning to solve networking problems, including routing, traffic engineering, resource allocation, and security. Recently, there has been a rising trend of employing machine learning to improve IoT applications and provide IoT services such as traffic engineering, network management, security, Internet traffic classification, and quality of service optimization. This survey paper focuses on providing an overview of the application of machine learning in the domain of IoT. We provide a comprehensive survey highlighting the recent progresses in machine learning techniques for IoT and describe various IoT applications. The application of machine learning for IoT enables users to obtain deep analytics and develop efficient intelligent IoT applications. This paper is different from the previously published survey papers in terms of focus, scope, and breadth; specifically, we have written this paper to emphasize the application of machine learning for IoT and the coverage of most recent advances. This paper has made an attempt to cover the major applications of machine learning for IoT and the relevant techniques, including traffic profiling, IoT device identification, security, edge computing infrastructure, network management and typical IoT applications. We also make a discussion on research challenges and open issues.

Keywords Machine learning · IoT · Networking · Application

1 Introduction

Internet of Things (IoT) is becoming a new pervasive and ubiquitous network paradigm offering distributed and transparent services [1]. Through IoT, lots of smart devices are connected, such as sensors, mobile phones and other smart devices. These smart devices can communicate with each other and exchange information. According to the IDC statistical report, there are over 50 billion IoT devices in the world; they will produce over 60ZB data by 2020 [2–4]. By collecting the data of these IoT devices and analyzing these data to sense and understand the environment, the complex systems can be constructed to enhance the quality of life, such as diagnosis of machine condition, human body activities, health monitoring, localization, and structural monitoring.

As the popularity and widespread use of IoT, the massive sensors and devices are generating massive data and various IoT applications are developed to provide more

✉ Fei Chen
fchen@szu.edu.cn

Laizhong Cui
cuilz@szu.edu.cn

Shu Yang
yang.shu@szu.edu.cn

Zhong Ming
mingz@szu.edu.cn

Nan Lu
lunan@szu.edu.cn

Jing Qin
harry.qin@polyu.edu.hk

[1] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, People's Republic of China

[2] Center for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Kowloon, Hong Kong

accurate and more fine-grained services to users. These IoT big data can be further processed and analyzed to provide intelligence for the IoT service providers and users. The emerging IoT applications involve many data-driven analytic procedures to efficiently utilize big IoT sensing data [5]. Recently the AI algorithms are introduced into the IoT data analytic procedures [6–8].

Over the past decade, the artificial intelligence (AI) achieves a great success with the advances in computing technologies of cloud computing, graphics processing unit(GPU) computing, and other hardware enhancements [9]. Machine learning is the most representative AI algorithm, which has been already applied in multiple fields, such as computer vision, computer graphics, natural language processing (NLP), speech recognition, decision-making, and intelligent control. Similarly, machine learning can also bring a potential benefit to computer network. Some researches studied how to utilize machine learning to solve networking problems, including routing, traffic engineering, resource allocation, and security [10–14]. Machine learning has been regarded as the key technology of autonomous smart/intelligent network management and operation. Especially, most IoT systems are becoming increasingly dynamic, heterogeneous, and complex; thus the management of such IoT systems is difficult. Moreover, the services of of such IoT systems need to be improved, in terms of effectiveness and diversity, in order to attract more users. A lot of studies have made progress on applying machine learning to IoT. Thus we can find that IoT can also benefit from leveraging support from machine learning. The application of machine learning for IoT enables users to obtain deep analytics and develop efficient intelligent IoT applications; this is because machine learning can provide feasible solutions to mine the information and features hidden in IoT data.

In this paper, we survey the application of machine learning for IoT by supporting the possible cooperation with use case scenarios. Meanwhile, we also study the current missing integration aspects of the machine learning and IoT for designating the challenges and future directions.

As a summary, the original contributions of this paper are as follows:

- We illustrate the potential of machine learning for traffic profiling. The unsupervised solutions and supervised solutions are presented detailedly.
- We make a summary on the IoT device identification with machine learning, in terms of mobile phone identification and general IoT device identification.
- We review IoT system security based on machine learning approaches, in terms of device security and network security.
- We summarize typical IoT applications leveraging machine learning, including personal health applications and industrial applications.
- We investigate the edge computing and SDN in IoT using machine learning, including edge computing infrastructure design and IoT network management.
- We also discuss the challenges and open issues on the reviewed areas, including traffic profiling, IoT device identification, security and edge computing, and SDN via machine learning.

The remainder of this paper is organized as follows. Section 2 introduces the progress about the application of machine learning to traffic profiling. Section 2 discusses how to use machine learning to identify IoT devices. Section 4 presents the security solutions of IoT systems by machine learning. Section 5 presents edge computing infrastructures based on machine learning. Section 6 describes how to use SDN with machine learning to manage the IoT network. Section 7 summarizes the typical IoT applications with machine learning. Finally, Sect. 8 concludes this paper.

2 Traffic profiling

Traffic profiling refers to the *fundamental* task of characterizing, understanding the traffic patterns in communication networks, including IP, wireless, mobile networks etc. It provides insightful information about the underlying traffic, thus helps manage, engineer the network to obtain better performance. For instance, among the benefits, detecting abnormal traffic specifically enhances the security of the underlying networks, which have gained considerable research efforts in recent years.

We define the traffic profiling problem as follows: the input of a traffic profiling task is the captured real network communication data; the output is a collection of patterns underlying the traffic. Figure 1 also shows the traffic profiling problem. Traditionally, researchers focused on investigating statistical properties of networks traffics, e.g., heavy hitters, heavy-tail, self-similarity [15–18]. While this approach obtains useful information for engineering networks, it is limited to particular networks. In recent years, researchers are leveraging the power of machine learning to profiling network traffics, which obtains more general results.

Here we review the progress of this area in the last decade, with a focus on security applications. We categorize the works into unsupervised and supervised solutions. We note that the categorization is based on whether background information is employed in the proposed solutions, which is different with traditional, theoretical, and abstract unsupervised/supervised learning; here we deal with domain-specific problems. We first summarize the core machine

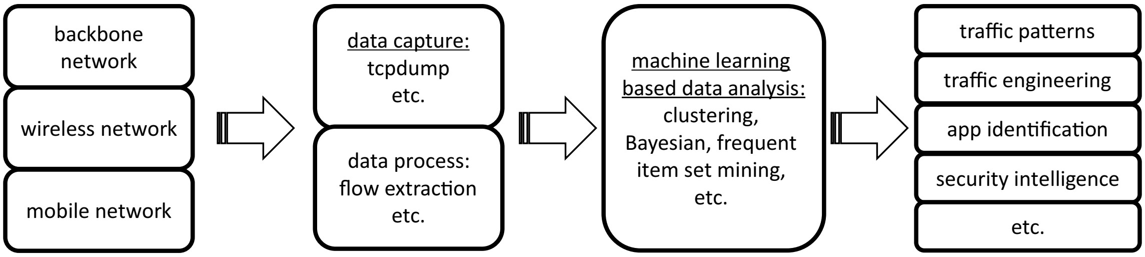
backbone network
wireless network
mobile network
data capture: tcpdump etc.
data process: flow extraction etc.
machine learning based data analysis: clustering, Bayesian, frequent item set mining, etc.
traffic patterns
traffic engineering
app identification
security intelligence
etc.

Fig. 1 Traffic profiling model

learning technique of the proposed approach, then present the detailed approach, and finally discuss the merits and limits. Table 1 lists a short summary of reviewed works.

2.1 Unsupervised solutions

Xu et al. [19] used the clustering technique to profile IP network traffics. The scheme first captures traffic data and aggregates the data into flows. Each flow has the same five dimensions: (`source IP, destination IP, source port, destination port, protocol type`). Next, the scheme clusters the data for each dimension, i.e. the source IP dimension, the destination dimension, etc. The significant clutters (according to the distributions) are output. We note that the significant clusters denote the patterns of the network traffic. The IP addresses information reveals the nodes' patterns of the communication traffic; the ports information shows the services patterns. Both contains important patterns of the traffic. The scheme employed a newly proposed entropy based metric to determine how may clusters are output. Then, for each cluster, the scheme analyzed the structures, i.e. similarities and dissimilarities, of the traffic. The scheme also studies how observed structure evolves with time. Based on the found structures, the scheme used dominant state analysis to model the interaction of the five dimensions (`source IP, destination IP, source port, destination port, protocol type`) in each cluster.

This scheme finally validated proposed approach on the core network traffic. The experimental results confirmed that the proposed scheme successfully found common, stable, and anomalous behaviors in the experiments. One noteworthy feature of the scheme is that the number of the clusters are adaptively determined. Along this line, the clustering technique is also used in [29] to profile higher level applications, specifically on Email. In future, one interesting, unknown problem is how different clustering algorithms influence the analysis.

Brauckhoff et al. [23] used frequent item set mining to detect anomalies for network traffics. The proposed scheme processes the captured traffic into seven-entry tuples (`srcIP, dstIP, srcPort, dstPort, protocol, #packets, #bytes`). The scheme first employs a traditional histogram-based detectors to filter out suspicious flows. For the filtered traffic, the scheme sets up a transaction with seven items (`srcIP, dstIP, srcPort, dstPort, protocol, #packets, #bytes`) for each suspicious flow. Then the scheme uses frequent item set mining to find the anomalies. For instance, if an IP address is flagged as an frequent item set, it may be an anomaly. The output of the proposed scheme are all the frequent item sets.

Table 1 Recent machine learning based traffic profiling works

Work	Problem	Technique	Data	Application	On security
2008 [19]	Traffic pattern analysis	Clustering	Core network	Traffic analysis	Median
2009 [20]	P2P identification	Frequent item set mining	Campus network	App. identification	Low
2010 [21]	Traffic identification	Cluster on graphs	Backbone network	App. identification	median
2011 [22]	P2P identification	General clustering	Backbone network	App. identification	Median
2012 [23]	Anomaly detection	Frequent item set mining	ISP	Intrusion detection	High
2013 [24]	Traffic classification	ML algorithm set	Campus	App. identification	Median
2014 [25]	Traffic visualization	Frequent item set mining	Campus	Visualization	Median
2015 [26]	Network user profiling	k means	Residential network	User profiling	Median
2016 [27]	Network user profiling	k means	DNS	User profiling	High
2017 [28]	User location prediction	ML algorithm set	Wireless AP	User profiling	Median

The scheme was validated on a median-sized ISP. First, ground truth is found using manual analysis, which is based to top-*k* queries. Then the scheme was used to identify anomalies. Experimental results show that the proposed scheme incurs a very small number of false positives. The biggest advantage of this scheme is that it reduces the time needed to analyze anomalies when detected. One challenging aspect of this approach is on parameter selection. In its current form, the threshold for frequent item set mining is by trial and error.

Glatz et al. [25] used frequent item set mining to profile network traffic and visualize the traffic. The proposed scheme first captures the traffic, obtaining a five tuple for each flow; flow statistics, e.g. payload size etc., may also be included. Then, the scheme employs frequent item set mining to find out top traffic flows. Later, the scheme plots the traffic using a hyper graph.

The scheme was validated on campus networks. By visualizing the traffic, it is easy to find the dominant traffic patterns, including popular network visits, network attacks, network misconfigurations, etc.

Bakhshi et al. [26] employed *K*-means clustering to profile network users into different behaviors, which is later used to engineering software defined networks (SDN). The proposed scheme first categorizes the captured network into different traffic types. This leads to a 9-entry tuple characterizing application layer services visits. The scheme then uses *K*-means clustering to group different user behaviors. The obtained user behavior is finally used to support software defined network designs.

The scheme is validated on a residual network. A commercial software NetFlow was employed to capture the traffics. When clustering, the scheme worked from 2 to 7 clusters in order to understand user behaviors. The idea of this scheme for SDN design is interesting.

2.2 Supervised solutions

Hu et al. [20] employed the frequent item set mining technique to identify P2P traffic from a bunch of network traffics. The main idea of the proposed scheme is to extract dominant, unique features from the P2P traffic using frequent item set mining. In order to get the features, the scheme first records P2P traffics delicately as trained data. The data is processed to contain the standard five tuple, i.e. (`source IP, destination IP, source port, destination port, protocol type`), and some manually statistical properties about the communication flow. Then the scheme uses frequent item set mining to find the the patterns that occur above a threshold. The patterns latter are later used to identify P2P traffic for online network traffics. It is worthy noting that a lot of engineering, scientific, heuristic efforts are done in order to obtain good results.

During performance evaluation of the proposed scheme, 10–15 patterns are obtained for BitTorrent traffics captured in a campus network. These extracted patterns prove to have more 90% accuracy when tested on real traffic. An interesting future work along this line is to adapt this approach to other approaches.

Iliofotou et al. [21] used clustering technique, specifically community mining in graphs, to profile traffics into different applications. The proposed scheme first uses the captured traffic to construct an IP-level connection graph; indeed, only IP addresses are used, but not depending on ports, payloads. Then the scheme finds clusters/commnities in the graph. For each cluster, the scheme tried to identify the underlying application using traditional signature analysis. The identified application is labeled for all the cluster, which is based on the intuition that the cluster shares the same application type.

Further performance evaluation on four backbone network traffics confirmed the effectiveness. The accuracy is around 90%. Besides, the scheme runs fast and works on encrypted traffic. An interesting future work could be lifting the accuracy by employing other useful information, e.g. port, payload.

Iliofotou et al. [22] combined clustering and statistical methods to profile network traffics, specifically on P2P traffics. The proposed scheme works on the traditional (`source IP, destination IP, source port, destination port, protocol type`) tuple. The scheme first groups captured traffic into similar flows using general clustering algorithms. Then for each cluster, the scheme generates a traffic dispersion graph. Leveraging statistical graph metrics of typical P2P traffics, the scheme determined whether a cluster is a p2P traffic.

The performance evaluation shows that the proposed scheme identifies 90% of P2P traffic in tested backbone networks. The accuracy achieves 95%. accuracy in backbone traces. In future, whether this approach adapts to other application types is worth studying.

Huang et al. [24] employed Naive Bayes, decision tree to classify network traffics into different high-level applications. The proposed scheme defined several statistics on early negotiation round of upper-layer application. Then taking these statistics as features and well captured knowtype flows, the scheme trained different classifiers for the traffic. The well-trained classifiers are later used for future traffic detection.

The proposed scheme was evaluated on campus traffics. Experimental results show that classifiers with the newly defined statistics have average 92% accuracy. Specifically, the accuracy is increased by around 7%, compared with the same classifier but without the newly defined feature. To employ this approach, one needs to know the total number of application types; for unknown traffics, how the scheme works is not known.

Kirchler et al. [27] used a *K*-means variant to profile network users according to the network traffic. The proposed scheme focused on DNS queries. The data is a vector showing how many times a flow visits a specific domain name; the dimension of the vector is the total number of different domains. The scheme then employs a modified *K*-means clustering algorithm to group different flows, which denote all the traffic of a specific user. Thus, the scheme successfully identified internet users.

The scheme is validated on a campus network DNS server. Among a period of 2 months, as high as 19% users were identified completely; 73% of the users in this subgroup can be linked over a period of 56 days. The accuracy is high. It is worthy noting that the scheme does not use traditional IP and port information. It is interesting to adapt this technique to other network traffics.

Das et al. [28] used a couple of machine learning technique to profile users in network traffics in order to identify user locations using traffic information only. The proposed scheme defined several flow level and application level statistics and used them as features to train machine learning algorithms. The ground truth is obtained by manually selection. The trained classifiers are later used to identify user locations.

The proposed scheme was validated on network traffics captured on wifi access points (AP). The highest accuracy is 89%, which is obtained by the Bayesian Network machine classifier. An advantage of this scheme is that it does not record user personal information, thus favors for user privacy. How to choose machine learning algorithms is also heuristic.

2.3 Challenges and open issues

Model reliability All the proposed schemes are valid on tested traffics. It is not known whether the model is still effective on different traffics in different ISPs, enterprises, countries, etc. One inherent reason is that traffic patterns change from time and space. In future, addressing reliability is both interesting and challenging.

Huge traffic volume Another challenge is how to deal large volume of traffic, both in storing them and processing them without lowering accuracy. Parallel machine learning algorithms or effective sampling may help. It is worthy further investigation in future.

Model security A very interesting and challenging problem is what if the traffic is perturbed with malicious flows. That is the input is not clean; a malicious adversary may purposely pollute the network traffic in order to fool classifiers. This needs to be further addressed in future.

3 IoT device identification

Device identification refers to a mechanism that predicts the type of an internet-of-thing (IoT) device according to the device's characteristics. Understanding the identifications of IoT devices is critical to service providers (e.g. mobile apps) for commercial purposes (e.g. advertising), and infrastructure (e.g. system/network) managers for security (e.g. finding vulnerable devices).

Specifically, we define the IoT device identification problem as follows: the input is various data collected from a device, e.g. sensors' data, network data, etc.; the output is a label for the IoT device indicating the type of the device. Figure 2 also shows the model for device identification. This problem receives extensive attention in recent year due to the proliferation of mobile computing, IoT depolyment, and smart everything. Since this area is rapidly evolving due to fast wireless and mobile technology innovations, we review recent efforts on leveraging machine learning to identify IoT devices in the last five years. Table 2 presents a short summary of the reviewed works.

It is worthy noting that proactive approaches are based on IP address, MAC address, unique device number by manufacturer or operating system are not stable; thus, researchers turned to machine learning approaches, which may also be passive identifications. In the following, we first review proposed approaches that tried to identify mobile phones, then we move to review works that aimed to identify general IoT devices.

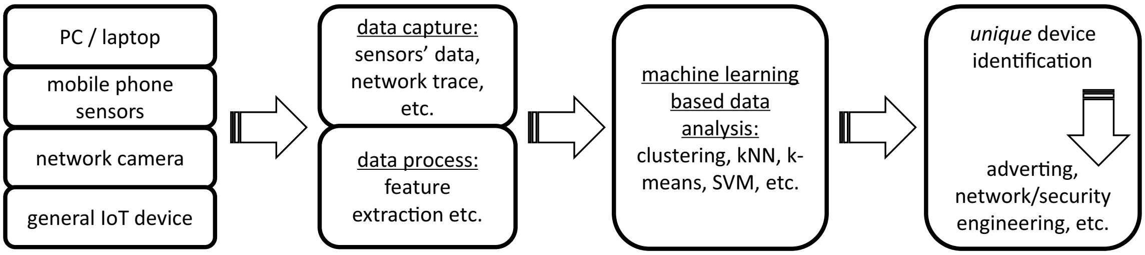
PC / laptop
mobile phone sensors
network camera
general IoT device
data capture: sensors' data, network trace, etc.
data process: feature extraction etc.
machine learning based data analysis: clustering, kNN, k-means, SVM, etc.
unique device identification
adverting, network/security engineering, etc.

Fig. 2 Device identification model

Table 2 Recent machine learning based device identification works

Work	Device	Technique	Data/signal	Accuracy (%)	On security
2013 [30]	Android	kNN, SVM	Network traffic	90	Median
2014 [31]	Android, iOS	kNN, GMM	Acoustic voice	98	Median
2014 [32]	Android, iOS	kNN, maximum likehood	Acoustic signal, accelerator	90, 50	Median
2015 [33]	ZigBee device	Decision tree, ensemble learning	Radio signal	90	High
2015 [34]	Android, iOS	kNN, GMM	Capacitive sensing	98	High
2016 [35]	Camera	SVM	Image signal	97	Low
2016 [36]	iOS	Threshold based classifier, SVM	Phone settings	90	High
2017[37]	Android, iOS	SVM	Magnetometer	94	Low
2017 [38]	Android, iOS	Random forestry	network traffic	95	High
2017 [39]	Android, iOS	General binary classifier	Network traffic	99	Median

3.1 Mobile phone identification

Stober et al. [30] used kNN and SVM to identify mobile phone based on network communication traffic. The proposed scheme records mobile traffic using tcddump for a time interval and further transforms the captured data into a 23-entry feature. The feature is heuristically defined based on traffic bursting patterns and not related to detailed payload contents. Then, the captured data is trained using kNN and SVM to identify mobile phones. For future phone identification, just apply the trained classifier.

The proposed scheme was validated on 20 Android mobile devices running 3G communication links. The accuracy achieves 90%. The computation is also efficient: the time needed to identify a mobile phone is about 15 min. The results indicate that mobile phones can be reliably identified/tracked even the communication traffic is encrypted.

Das et al. [31] employed kNN and Gassian mixture model to identify mobile phones based on acoustic data. The proposed scheme captured and extracted features on acoustic signals from the microphone and the microspeaker of a phone. In total, 25 features are defined. Then the proposed scheme trained machine learning models (i.e. Gassian mixture) on the captured data. Later, the trained model or kNN is used to identify mobile phones.

The proposed scheme was validated in lab on 52 mobile phone. Both iOS and Android systems exist. Experimental results showed that devices manufactured by different vendors can be effectively identified. In addition, devices from the same manufacturer and model can also be identified. The accuracy is as high as 98%.

Bojinov et al. [32] used kNN and maximum-likelihood classification to identify mobile phones based on acoustic/acceleratormeter sensors. The scheme first tried to identify features for acoustic/accelerator sensors. The features reflect a basic fact that each sensor is imperfectly manufactured and thus has its unique noises. Based on the unique features, the scheme just uses simple kNN to identify mobile phones.

The proposed scheme was tested both for the acoustic sensor (i.e. microphone and microspeaker) and the accelerator sensors. For the former, a 90% high accuracy is achieved for the former. For the latter, a 50% accuracy is obtained with the additional user-agent string for web browsing. An interesting future work along this line is to employ more powerful machine learning algorithms and more features to pursue higher accuracy. Currently, only few features are used.

Huynh et al. [34] used kNN and Gaussian mixture to identify mobile phones with the touch screen sensor. The proposed scheme is based on the fact that every touch screen of mobile phones are different. The scheme employed 16 different features regarding to signals generated by capacitive sensing. Then Gaussin mixture model and kNN are used to identify mobile devices.

The proposed scheme was tested on 14 mobile phones with Android, iOS as operating systems. The identification accuracy achieve 98%. Such a high accuracy has various potential applications for authentication scenarios, e.g. ATM authentication, smart unlocking, as pointed out in the paper [34].

Kurtz et al. [36] used threshold based classifier and SVM to identify mobile phones for Apple's iOS system. The proposed scheme employs manually defined feature from phone settings, including public (device model, the current iOS version, etc.) and protected resources (location data, photos, contacts, calendar data, reminders, sensor data) of the phone. In total, 29 features are defined. The scheme then tested the effectiveness of these 29 features. A threshold based classifier and a linear SVM classifier are trained and employed to identify phones.

The proposed scheme was implemented as an iOS app and tested. For the threshold based classifier, more than 90% accuracy was achieved; for the SVM classifier, roughly a little higher accuracy was obtained, but with the added more computation overhead. An interesting future work is to adapt the proposed scheme to Android and other IoT devices.

Baldini et al. [37] employed SVM to identify mobile phones based on magnetometer sensor. The proposed scheme first captured the magnetometer digital output with a given sampling time using an app in the phone. The scheme then extracts Shannon entropy, log energy entropy, variance, standard deviation, skewness, and Kurtosis from the captured data as features. The features are then input an SVM to train the later classify mobile phones.

The proposed scheme was validated on ten phones of different of different brands and models. The classification accuracy achieved 94% for mobile phones of different brands and models. However, intra-model classification has an accuracy around 54%.

3.2 General IoT device identification

Patel et al. [33] employed decision tree combined with random fores and with multi-class Ada boost to identify ZigBee devices. The proposed scheme first defined statistical features on radio signals, including signal's instantaneous amplitude, phase, frequency, ect. Then the scheme collected the features for known devices and trained decision tree models. For device identification, just capture the features for the device and input the features into the classifiers.

The proposed scheme was validated. The accuracy can achieve 90%. This scheme is also capable of detecting unknown ZigBee devices in a given networked system. One interesting future work is to enhance the feature space and check whether accuracy could be improved.

Tuama et al. [35] employed support vector machine (SVM) to identify cameras according to images. The proposed scheme leverages the detailed photo-taking process of cameras, and then deduces features of a camera. Specifically, more than 10,000 features on co-occurrences matrix, color dependencies, and conditional probability of an image are used. The scheme then trains a SVM model to classify different images to different cameras. The trained model is later used to identify cameras.

The proposed scheme was validated on a public image database. An SVM based on radial basis function (RBF) was trained on 100 images and later tested on another 100 images. Experimental results showed that the identification accuracy achieves more than 97%.

Miettinen et al. [38] used random forest to identify general IoT devices e.g. smart lighting, home automation, security cameras, household appliances and health monitoring devices. The proposed scheme employs the network communication data to extract features. Specifically, 23 features of first a few network packets during initial communications are employed. The scheme then trained a classifier model for each IoT device using the captured data on the 23 features. Finally, the trained models were used to identify IoT devices in the network fastly.

The proposed scheme was tested on a representative set of consumer IoT devices in the European market. A data set of 540 fingerprints representing 27 device types was obtained for training and validation. Experimental results showed that the accuracy of identification achieved over 95% for 17 devices, and around 0.5% for the remaining 10 devices with the same manufacturers. As shown by the paper, the device identification results can be further used by SDN controllers to enforce security policies in the IoT network composing of various devices. An interesting future work along this line could be using other machine learning models and more features to increase identification accuracy.

Meidan et al. [39] employed binary classifiers to identify general IoT devices, e.g. smart TV, IP camera, baby monitor, etc. The proposed scheme works on network traffics between the IoT devices and access points. Specifically, various packet and payload characteristics are used to extract features. The scheme then trains a binary classifier for each IoT device using captured network traffic. When identifying a device, each model is used for multiple network sessions. Finally, a majority vote is used to determine the exact device.

The proposed scheme was validated in a local wireless network environment with multiple IoT devices, including PC, smart phone, a few sensors. Experimental results showed that more than 99% accuracy were obtained.

3.3 Challenges and open issues

Defense approaches One interesting future direction is on the interplay of IoT device identification and defense. Traffic encryption/padding, false traffic injection, and mobile phone OS priority protection are potential defense strategies. How to defend device identification and how to identify devices with protection mechanisms are important research directions.

Understand the efforts of different machine learning approaches Another interesting problem relates to how to choose machine learning algorithms. Researchers have proposed a bunch of algorithms. How different algorithms influence IoT device identification and how to define features input to the machine learning algorithms are yet to be understood.

Privacy evaluation Further, all reviewed work here show that IoT devices can be identified with high accuracy. This breaks user privacy. A deep privacy evaluation and potential protecting methods are worthy studying.

4 Security

Security problems in IoT networks are more and more important with the increasing number of attacks nowadays. The IoT networks are more vulnerable than traditional network because of the characteristics of IoT devices and communication protocols. For example, (1) IoT devices are usually equipped with lower battery and micro-controller, thus it is easy to be flooded; (2) IoT communicate with each through Bluetooth, ZigBee, WIFI or GSM, which are more vulnerable to attacks.

In an IoT network, there are usually three components, including devices, gateway and controller. All of these components could become targets of potential attackers. Figure 3 shows the IoT security problem model. In the model, traffic, wireless signals, device events and configuration files could be analyzed. With the feature extracted from the data sources, varieties of machine learning methods are used to classify the data. The results can be used for privacy and authentication, or judging whether an event belongs to intrusion or anomaly events.

Here we review the progress of this area. We categorize the works into device security which focuses problems on an isolated device, and network security which focuses problems across the whole network. The network security mechanism could be installed on gateway or controller, depending on the specific environment. Table 3 lists a short summary of reviewed works.

4.1 Device security

Kotenko et al. [40] used a combination of multilayered perceptron and probabilistic neural network to forecast the state of an IoT element. The paper used the volume of the traffic, the rate of service, the rate of losses, the length of packets queue and time of user action to inquiry as indicators, to unambiguously define the state of an element. The solution is expected to reduce of cost of administration, especially during emergency.

The proposed solution was evaluated in MathCAD. The results show that the combined ANN model can provide higher precision. Compared with recurrent ANN model, the convergence of the combined model was close to 100%, where the recurrent model fell to 15% in certain cases.

Baldini et al. [41] used RF fingerprinting for device authentication, which was computed based on permutation entropy and dispersion entropy for cellular communication devices, including WIFI, GSM, etc. The mechanism is not easy to crack, because it is based on the physical properties of cellular devices. The authors in the paper also make a comparison between different machine learning classification methods, including KNN, SVM and decision tree.

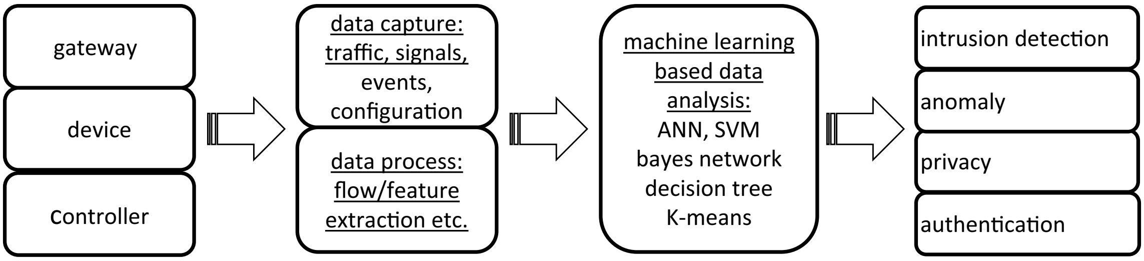
gateway
device
controller
data capture: traffic, signals, events, configuration
data process: flow/feature extraction etc.
machine learning based data analysis: ANN, SVM bayes network decision tree K-means
intrusion detection
anomaly
privacy
authentication

Fig. 3 IoT security model

Table 3 Recent machine learning based security mechanism works

Work	Problem	Technique	Data/signal	Accuracy (%)	Security granularity
2015 [40]	Security of an IoT element	Combined ANN	Device traffic	100	Device
2017 [41]	Authentication of an IoT element	KNN/SVM/decision tree	RF signals	80	Device
2017 [42]	Authentication of an IoT element	None	RF signals and environment parameter	None	Device
2017 [43]	Privacy of an IoT element	NN	Device traffic	None	Device
2015 [44]	Security of an IoT element	Bayes Algorithm	Device properties	None	Device
2016 [45]	Intrusion detection	SVM	Device traffic and events	100	Network
2016 [46]	Security of IoT networks	ANN	Device traffic and events	99	Network
2016 [47]	Security of Mobile networks	SVM/NN	Signals and environment parameter	None	Network
2013 [48]	Intrusion detection	SVM/Bayes network/decision tree	IDS events	50–78	Network
2013 [49]	Intrusion detection	SVM/K-means	Device traffic and events	None	Network

The authors used a set of 9 nRF24LU1+ wireless devices, where the RF signals are transmitted. The signals were collected by a software defined radio (SDR). The results show that the overall accuracy can achieve 80%, which is enough to support multi-authentication wireless IoT devices.

Sharaf-Dabbagh et al. [42] presented a demo, which also uses RF fingerprinting for wireless device authentication. Additionally, the proposed framework monitors the noises of communication channels and the environment surrounding the source object. Hence, the fingerprinting is more robust compared with previous solutions. The authors also setup a demo which consists of multiple Raspberry Pi boards.

Jeong et al. [43] proposed a new approach that can protect user privacy when running cloud based machine learning algorithms. Traditionally, the cloud collects user data, which is sensitive and vulnerable. The new approach let the clients compute the partially-processed feature data obtained from the early state of neural networks, and the server continues the rest stages after receiving the feature data. Thus, the service is safer, while the data in transmission is not easy for reverse-engineer. The authors also measured the performance with a testbed, where an embedded board equipped with ARM big.LITTLE CPU acted as the client, and a desktop PC equipped with x86 CPU acted as the server. The results show that the new approach have a shorter prediction time while improving the privacy.

Jincy et al. [44] created a general security framework for IoT devices. Due to the increasing use of variety of IoT devices, currently we do not own a specific security mechanism which adapts to all kinds of IoT devices. the framework in this paper classifies the devices to different types indicating the capability to support security mechanism, based on their capabilities and parameters, such as power, processing, scalability, network layer, etc. The authors also found that naive bayes algorithm is appropriate for the above purpose. Thus after inputing a file consists of listed properties, the system will output the class of the device, e.g., Class A for Critical, Class B for Medium and Class C for Non critical devices.

4.2 Network security

Nobakht et al. [45] proposed a intrusion detection and mitigation system for smart home called IoT-IDM. The proposed scheme collects traffic and events data from various devices in a smart home. The data will be transported to the SDN controller, where IoT–IDM is deployed. At last, IoT–IDM use linear regression model and SVMs to obtain the optimal classification model.

The proposed scheme was tested with an experiment setup where Philips lighting system is employed, and a realistic setting is used. The accuracy of the linear regression model was 96.2%, and the accuracy of the SVMs model was 100%.

Canedo et al. [46] deployed machine learning within an IoT gateway, to address the challenges, including heterogeneity and quantity of devices across an IoT network. The edge devices in the network will collect data and transfer them to the gateway devices. The gateway devices will use artificial neural networks (ANN) to learn the healthy state of each device and make informed decisions.

The proposed solution created an IoT testbed, where Arduino devices were used to simulate the edge devices, and Raspberry Pi devices were used to simulate the gateway devices. The simulation results show that ANN can make correct prediction over 99% of the time.

Do et al. [47] proposed a mobile security system using machine learning. The new system can be used to improve vulnerabilities in mobile networks, such as phone and IoT networks. Compared with previous systems, the machine learning based system can better solve security problems, including zero day attacks and construction of conclusive attack signatures. The authors also presented a case study about Man-In-The-Middle attack with IMSI catcher. In the case study, SVN and neural networks are used to detect the anomalies.

Stroeh et al. [48] proposed a security mechanism, which does not rely on network traffic, but correlates the attacks with security events or alerts provides by sensors, such as IDSs, logs, etc. The system first collects raw data and transforms them into standardized format. Secondly, the system take the standardized alerts and cluster them into meta-alerts, which contains a structure called alert_taxonomy_set, a bit array that represents each of the supported alert types. At last, the system will sort the meta-alerts into attacks and false alarms.

The authors implemented and tested the new systems against two major data sources, including DARPA challenge and SotM from the honeynet project. Three different machine learning techniques, including SVM, Bayesian Network and decision tree, are used. The results show that the detection rate increase from 40–60% to 50–78% with different operating systems and attack types.

Rathore et al. [49] proposed a bio-inspired machine learning mechanism for improving wireless sensor network security. To address the challenges faced by current wireless sensor networks, with increasing number of nodes and complexities of network topologies. The authors were inspired by the human immune system, which have intelligent capabilities of detecting anomalies in the body. The system first classifies the nodes into fraudulent or benevolent nodes. According to the classification results, the system will generate virtual antibodies, which in advance will have an impact on the trust rate. Finally, the gateway will make a decision whether or not to attack the fraudulent nodes.

During the classification phase, SVM and K-means algorithm could be used.

4.3 Challenges and open issues

DDoS attack Unlike previous attack detection system, machine learning based system consumes more computing resources. Thus, itself becomes the choke point, which is vulnerable to DDoS attack. How to make a tradeoff between accuracy and computing complexity is an important research direction for machine learning based security mechanism.

Security infrastructure Lots of works are devoted to intrusion and anomaly detection for IoT network/elements. There are much room for research on detecting attacks targeting at the security infrastructure and key distribution mechanism.

Data acquisition Almost all work use traffic, event, and signal data, for security analysis. However, for some attack, it is hard to detect using these data. At the same time, there are large volume of security related data in the IoT network, including administration, configuration, and routing data, etc. It is worth investigation on how to make better classification decisions with these data.

5 Edge computing with machine learning

In the IoT world, sensors and equipments are all around the network, including the edge network. Lot of IoT applications put requirements for latency, bandwidth, security on the network, and cloud computing can not satisfy such requirements. Edge computing is a promising new technology that can satisfy such demands [50]. For example, (1) VR and AR applications that needs high bandwidth can fetch contents from the edge network; (2) Vehicles can exchange data with each other through the edge networks, supporting vehicles on roads to act co-operatively and providing better user experience [51]. In the following sections, we use "edge computing" and "fog network" interchangeably for convenience.

Figure 4 shows the edge computing problem model in the IoT networks. In the model, traffic and sensor data could be analyzed. With the feature extracted from the data sources, varieties of machine learning methods are used to classify the data. The results can be used for intrusion detection, image recognition, diseases identification, traffic engineering, etc. Table 4 lists a short summary of reviewed works.

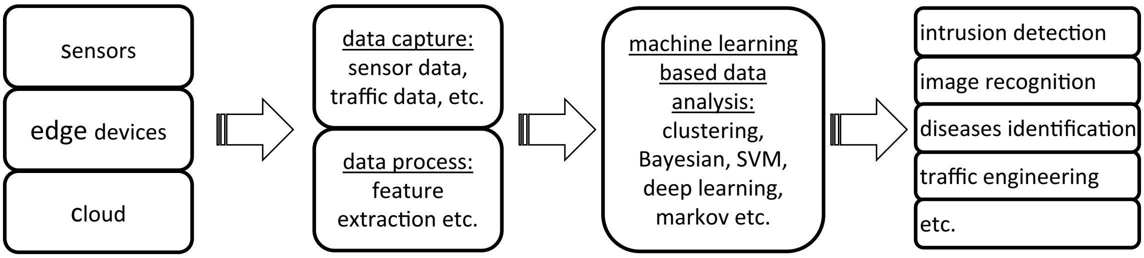

Fig. 4 Edge computing in IoT network model

Table 4 Recent machine learning based edge computing works

Work	Problem	Technique	Data/signal	Accuracy (%)
2017 [52]	Identification of Parkinson's disease in edge computing	Clustering	Speech data	None
2017 [53]	Image recognition in edge networks	Markov-model	Images	90
2017 [54]	Arrhythmia detection	Linear support vector machine	ECG of patients	93.6
2015 [55]	Evaluating parking availability	Cascade classifier	Video captured by smartphone	None
2017 [56]	Service recommendation in mobile edge computing	Collaborative filtering	User mobility information	64.4
2017 [57]	Anomalies detection	Support vector machine	Sensor data	90
2017 [58]	Anomalies detection	Federated learning	Sensor data	95–98
2018 [59]	Distributed attack detection	Deep learning	Traffic data	92
2018 [60]	Privacy protection during data aggregation	Linear regression algorithm	Sensor data	90
2017 [61]	Traffic engineering	Bayesian Networks	Path latency traces	80–90

5.1 Edge computing applications

Borthakur et al. [52] proposed a new framework in a smart telehealth system with kinds of wearable devices. In the framework, the authors suggested the use of the edge computing devices, that have lower resources, but locate more closely to the end user. Firstly, the paper described a new architecture for telehealth computing such that decentralization of services at the edge network can be achieved. Secondly, algorithms of speech signal recognition are used for telehealth monitoring, and K-means clustering is used to identify parkinson disease.

Drolia et al. [53] proposed a system called Precog, which accelerate image recognition through enabling caching and prefetching on the edge devices. The system is collaborated between three parts: devices, edge server and cloud server. Unlike previous edge computing solutions that all computing tasks could be completed on the edge server, Precog also uses computing resource on devices and cloud server, due to the computing complexity and data volumn of image recognition. Both the edge server and device will employ recognition cache that stores relevant parts of the trained model. What is more, devices will prefetch part of the trained classifiers which are predicated to be used in the near future.

Azimi et al. [54] proposed a hierarchical computing architecture (HiCH) for healthcare IoT network. In the architecture, the existing machine learning methods are partitioned among different layers of the fog network. For example, the sensor devices are responsible for sensing and monitoring; the edge computing devices are responsible for local decision making and system management; and clould is responsible for heavy training procedures. The authors devise a system based on IBM's MAPE-K model, and demonstrate a complete implementation that focus arrhythmia detection. The results show that HiCH out-performs traditional systems in both response time, bandwidth utilization and storage, while the accuracy is acceptable.

Grassi et al. [55] devised a low-cost crowdsourcing architecture called ParkMaster in visual analytics for evaluating parking availability. Different with traditional centralized monitoring system, ParkMaster makes use of smartphones, which locates inside the car, captures the video stream along the street and count the number of detected cars after processing the video with machine learning methods. The processing results are uploaded to the ParkMaster cloud, that processes data from multiple cars and recommends a parking slot for a driver.

Wang et al. [56] proposed a service recommendation system based on QoS prediction in mobile edge computing environment. Unlike other context-aware service recommendation system, the proposed system takes mobility into consideration. Based on the mobility information, the system recommend service to users by using collaborative filtering algorithms. The authors carried out a series of experiments based on the data from Shanghai Telecom, and the results show that the new system can achieve higher prediction accuracy.

5.2 Improving edge computing infrastructure based on machine learning

Zissis [57] presented an intelligent intrusion detection system (IDS), which can secure the underlying edge computing infrastructure. The system improves the traditional "Self-protecting" system proposed by IBM. By collecting data from sensors, and making use of the latest unsupervised machine learning method, the system can intelligently detect the abnormal devices which could be harmful to the whole system. Finally, the author developed a proof of concept system, that can detect anomalies in real world.

Schneible et al. [58] also presented an anomalies detection system in edge computing environments using artificial neural networks. In traditional neural networks, all sensor data and the training model should be placed in one place, i.e., the centralized cloud. This characteristic brings congestion and latencies along the traveling path. The new system is based on federated learning, where training data is split among the edge devices, and each edge device stores a copy of the training model. Thus, the centralized cloud repository only needs to aggregate the training results from edge devices. The federated learning mechanism can improve latency, bandwidth and make full utilization of computation power across the edge networks.

Abeshu et al. [59] further investigated the distributed attack detection problem, which is more difficult to detect compared with non-distributed attack. In the edge computing environments, due to the diversity and complexity of devices, traditional machine learning mechanisms have low accuracy and less scalability. Thus, the authors proposed a new scheme based on deep learning, which is popular in recent years due to the advancement of GPU hardware and theory in deep neural networks. At last, the authors evaluated the new mechanism through simulations based on publicly available datasets. The results show that the DL based mechanism outperform the traditional methods.

Besides security considerations. privacy is also an important aspect for fog computing environment. Yang et al. [60] proposed a machine learning based privacy protection mechanism when devices aggregate data from sensors in a fog computing architecture. More importantly, the new mechanism supports multifunctional data aggregation method, thus it can support a wide range of data sources. The system also distributes the computationally heavy tasks to the edge of the network, making the system more scalable then centralized system. The experiment results show that the system achieve high accuracy without disclosing user privacy.

Hogan et al. [61] proposed a solution for traffic engineering in edge networks. In the edge networks, there may exist multiple end-to-end paths, where each path has different delay and bandwidth. Choosing one that bestly match requirements of users is especially important. The solution mentioned in this paper computes the results based on portfolio theory, which maximizes the expected return given the level of risk (representing the expected throughput during the lifetime). Given the model, the authors use machine learning to evaluate the level of risk for each path. At last, using real-world latency traces, the paper compare the proposed solution with other techniques, and the results show that the solution leads to better performance.

5.3 Challenges and open issues

Heterogeneous data types In the edge network environment, the data sources are composed of heterogeneous sensors. The collected information could contain a diversity of data types, even contain uncertainty under some circumstance. Some data may be incomplete, which add extra complexity to the system. Thus, new architecture in edge computing should take this heterogeneous into consideration.

DDoS attack on edge devices Lot of work has been devoted to DDoS attack on cloud computing, which has tremendous capacity and is well designed to defend such attack. In the edge computing, (1) the edge devices have lower capacity; (2) the infrastructure is not so mature compared to traditional cloud computing. Thus, the edge computing environment is more vulnerable to DDoS attack, especially when the attackers desire to flood a specific attacking point.

Convergence speed for distributed machine learning Previous work proposed to use distributed machine learning in the edge computing, where sensors and edge devices is responsible for local and light-weight training procedure, and centralized cloud is responsible for global and heavy training procedure. The distributed machined learning mechanism can improve latency time and achieve similar training accuracy. However, previous work has not taken convergence time into consideration, due to the transmit time between local and global training point could become a bottleneck of the whole system.

6 Software-defined networking with machine learning in IoT

Recently, both academic and industrial fields have seen emerging of software-defined networking (SDN) due to its flexibility. SDN separates control plane from forwarding plane, thus network operators can manipulate the network with high level configuration language, and do not need to take the complex forwarding table configuration into consideration. Due to the complexity and diversity of IoT devices, data path configuration in IoT network is even more difficult compared with traditional network. Thus, SDN can play an important role in the IoT world [62]. However, also due to the complexity of IoT, the control plane needs machine learning for better management of the networks.

Figure 5 shows the software-defined networking problem model. In the model, traffic and sensor data could be analyzed. With the feature and flow extracted from the data sources, varieties of machine learning methods are used to classify the data. The results can be used for intrusion detection, traffic management, fault detection, DDoS attack detection, etc.

In this section, we will investigate into the previous works related with machine learning for SDN in IoT network. We will review two important aspects, (1) how machine learning can IoT network management with SDN more easily and effectively; (2) how machine learning can help detect the possible intrusion and increase the accuracy.

6.1 IoT network management

Kim et al. [63] proposed a new solution that can identify the service context of a flow, and infer the QoS requirements of the flow. Because SDN will assign a flow to a virtual network, thus the service context identification is important for flow assignment and virtual network construction. However, the service context identification is not straightforward, i.e.,

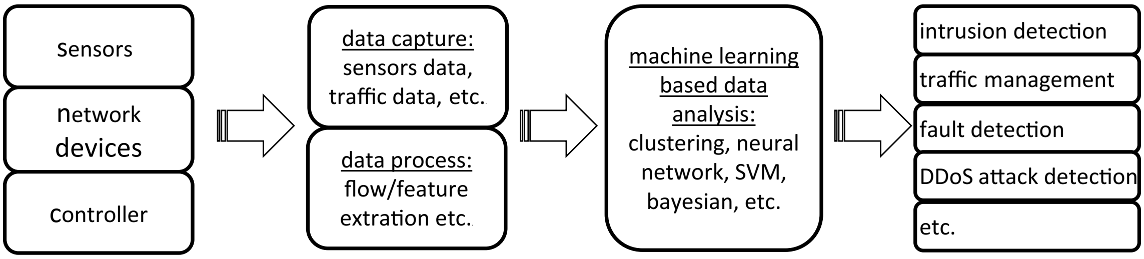
Sensors
network devices
controller
data capture: sensors data, traffic data, etc.
data process: flow/feature extration etc.
machine learning based data analysis: clustering, neural network, SVM, bayesian, etc.
intrusion detection
traffic management
fault detection
DDoS attack detection
etc.

Fig. 5 Software-defined networking in IoT network model

it can not be derived directly from the protocol field, because the packets could be encrypted. Thus, the authors proposed that machine learning could be used to classify the flows, based on their characteristics, such as mean packet length and mean inter-arrival time, etc.

Vukobratovic et al. [64] presented a novel architecture called Condense which integrate data analysis function into the IoT infrastructure, thus data manipulation such as aggregation and computation, could be done along the path. With Condense, data redundancies could be reduced, and network bandwidth could be saved, in the IoT networks. To implement Condense, the authors proposed a function computation interface, which should be placed between data communication and analysis. SDN can be used for implementation of the function computation. With this enhancement, machine learning tasks can also be integrated into the Condense architecture, that is, the learning tasks could be seen as a series of function computation across the network (Table 5).

Jagadeesan et al. [65] described a new mechanism for software faults detection in software-defined IoT network. Although SDN brings flexibility in the control plane, and operators can control the network through high level language such as JAVA or python, it could make the network more vulnerable to software faults. Considering the complexity and heterogeneity of the IoT network, the problem could be even more serious. The authors in this paper proposed to use machine learning to classify the encountered problem into software faults and other problems.

Taneja [66] proposed a framework for traffic management in IoT networks using SDN. In the IoT networks, traffic management is more necessary and more difficult, because of the great differences between different devices. Fortunately, most IoT communication protocols support traffic classification. For example, 802.11ah can classify devices into TIM (Traffic Indication Map) devices and non-Tim devices, and LoRaWAN can classify devices into class A, B or C. The authors put forward a new management mechanism, where SDN is used to perform dynamic management of traffic class, and transmit requirements prediction in the near future. Thus, machine learning could be used to increase the accuracy of prediction.

6.2 Intrusion detection

Nobakht et al. [45] proposed a host-based intrusion detection system (IDS) based on openflow for smart home environment. In the new system, the controller collects and analyze data from sensors. Using machine learning, the controller can judge whether there is intrusion or not. The authors also implemented a proof of concept system called IoT-IDM based on Floodlight. Machine learning methods could be used as a module in the system. The paper also studies a special case in a smart light system, and the results show that the system can bring flexibility with SDN, and achieve high accuracy with appropriate machine learning algorithm.

Uwagbol et al. [67] presented a pattern driven corpus to predict SQL injection attack. Although SQL injection attack is well studies, the problem arises again because IoT and SDN networks bring new opportunities for the attackers, and the defenders lack a readiness corpus for machine learning method that could identify new attacks. The authors in this paper presented a pattern driven corpus generation mechanism based on finite state automata. With the generated corpus, machine learning methods could be used to train a new model. Finally, two publicly datasets are used to evaluate the accuracy of the proposed mechanism, and the results show that the mechanism can achieve high accuracy.

Most previous work tried to secure DDoS attack in the IoT world, Ahmed et al. [68] made use of machine learning methods to identify DNS query-based attack. Unlike DDoS attack, DNS query-based attack could be launched with only a small number of packets, thus it would be more harmful if used. In the proposed system, the SDN controller will collect traffic data from the network, and identify the DNS query-based attack traffic based on machine learning. At last, the authors implement a prototype based on dirichlet process mixture model, and conduct simulation based on real-world traces. The simulation results show that the machine

Table 5 Recent machine learning based SDN in IoT networks

Work	Problem	Technique	Data/signal	Accuracy (%)
2017 [63]	Identification of service context	Clustering	Traffic data	None
2016 [64]	Distributed learning with SDN	Neural networks	Device data	None
2016 [65]	Software faults detection	Clustering	Packets data	None
2016 [66]	Traffic management	None	Traffic data	None
2016 [45]	Intrusion detection in smart home	SVM & Logistic aggregation	Sensor data	96.2
2017 [67]	SQL injection attack detection	SVM	Traffic data	96.4
2017 [68]	DNS query-based attack detection	Bayesian	Traffic data	75
2017 [69]	Dynamic attack detection	SVM	Traffic data	98

learning based method outperform traditional mean shift based method.

Bhunia et al. [69] proposed a dynamic attack detection system called SoftThings, which is based on SDN, and tries to prevent attacks at the network layer rather than device layers. Thus, the network can eliminate the malicious traffic as early as possible. The system system is divided into three layers: devices layer, cluster SDN controller and centralized master controller. The distributed SDN controller will monitor and detect anomalous behaviors of IoT devices. Once found, the mis-behavior will be reported to the master controller, which will dynamically judge whether there exists attacking or not. Finally, the authors conducted simulations on Mininet emulator, and the results show that SoftThings greatly improve the performance of traditional attack detection system.

6.3 Challenges and open issues

Overheads in control plane Overload in control plane is a possible problem in SDN network. In the IoT environment, the overheads could be even higher with huge number of data sources and high complexity of machine learning methods. Thus, it is important to guarantee that the additional overheads will not crash the system.

DDoS attack on controllers As mentioned above, the overheads in control plane will be quite high. It will become a vulnerable point in the system. Once the attackers find the patter that can bring large overheads to the system, in both data collection and training procedure. It can make the whole system collapse with high probability.

Controller placement In the IoT world, controller could be place almost everywhere, e.g., edge network, centralized cloud based controller. However, different placement mechanism bring different performance. For example, controller on the edge can reduce the communication latencies, and controllers on the centralized cloud can bring large computation capacity. Depending on the application scenario, controller placement should be well studies to satisfy user demands.

7 IoT applications

In recent years, IoT applications are constantly emerging in almost all fields, e.g., health, agriculture, industrial, etc. However, IoT applications are facing great challenges, due to the heterogeneity and complexity of the data sources. In Fig. 6, we show the model of IoT applications. In the model, the potential data sources include wearable device, mobile phone sensors (like accelerometers), network camera and kinds of wireless sensors. These sensors capture human vital signs like temperature, ECG, and environment data like humidity, meter readings and camera images. Then, varieties of machine learning methods could be used for human health monitoring, human activity recognition, fraud action detection and object detection.

There are lots of works on IoT applications. We mainly review the previous works on personal health monitoring and industrial applications. For example, IoT networks can be used to monitor human stress, recognize human activity or presence in the personal health field; or predict agriculture disease and fraud action in smart cities. Table 6 list a short summary of review works.

7.1 Personal health applications

Asthana et al. [70] presented a recommendation system that advises wearable IoT solutions and wearable devices for any individual. The system first collects available user health data, including health history, demographic features and previous collected IoT data from medical or health sensors. Then, with classification models like decision tree, logistic regression and LibSVM, the system makes predications about the diseases. Each disease is related to some attributes that need to be monitored. At last, a mathematical optimization model is used to recommend the best IoT solution or wearable devices.

Walinjkar et al. [71] proposed a prognostic approach based on real-time Electrocardiograph (ECG) analysis. With real-time data from constantly ECG monitoring devices, the scheme first analyze the ECG waveforms with K-NN or

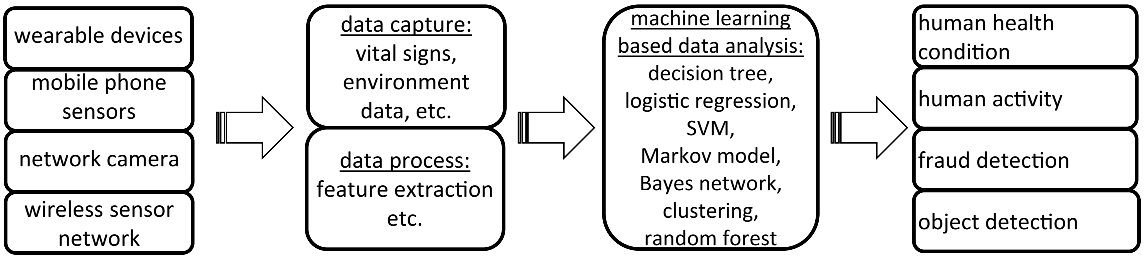

Fig. 6 IoT application model

Table 6 Recent machine learning based IoT application works

Work	Problem	Technique	Data/signal	Accuracy (%)
2017 [70]	Recommend IoT solutions and wearable devices	Decision tree, logistic regression, LibSVM	Health history	None
2017 [71]	Prognostic based on ECG	Bagged tree, K-NN	ECG waveform	99.4
2017 [72]	IoT health architecture	None	Sensor data	None
2016 [73]	Human presence detection	C4.5 decision tree, LinearSVC, random forest	IoT devices	50–90
2017 [74]	Human stress detection	SVM, logistic regression	Pulse waveform	68
2011 [75]	Human activity recognition	Logistic regression, multilayer perceptron	Accelerometer data	90
2016 [76]	Grape disease prediction	Hidden markov model	Environment data in yards	90.9
2017 [77]	Smart meter operation	Bayesian network, naive bayes, decision tree, random forest	Smart meter data	96.69
2017 [78]	Parking space detection	Clustering algorithm	Camera data	97
2015 [79]	Flowering dynamics in rice	SVM	Camera data	80

other classifier. The system can predict arrhythmia and other ECG abnormalities. The authors also setup a monitoring IoT network, where the analysis results will be transferred to NHS (National Health Services, UK) cloud in real-time. In this paper, two classifiers including bagged tree and K-NN were used, the test results show that the precision can reach 99.4% if K-NN is used.

Nguyen et al. [72] explored the IoT application in medical field and proposed an IoT tiered architecture, which collects sensor data, analyze them and transform them into clinical feedback. The architecture is divided into five layers: (1) sensing layer, which uses sensors, actuators and wearable devices to gather data; (2) sending layer, where kinds of communication mechanism including Wifi, Bluetooth, ZigBee and LTE could be used to send the data to the cloud; (3) processing layer, which could happen on smart phones, micro-controllers and micro-processors. Notifications and alerts could be generated if necessary after processing; 4) storing layer, where data can be stored in clouded or hosted servers; 5) mining and learning layer, which converts information to decisions or predictions using mining or machine learning algorithms.

Madeira et al. [73] described a system that can detect the human presence using IoT devices, and do not rely on devices, like cameras and motion detectors, that explicitly detect human presence. The system first collects interaction data, e.g., reading and writing, with the large diversities of devices. Then using machine learning algorithms, the system can predict the human presence. The system was tested using a dataset gathered during 3 days from 900 users. The authors also tested a set of classification methods, including C4.5 decision tree, LinearSVC and random forest, to make prediction. The results show that the precision ranges from 50 to 99% according to the algorithm selection.

Pandey [74] used individual heart beat to predict whether a person is in stress or not in an IoT network. The author designed a Wifi-equipped board that can detect pulse waveform. The board will transfer the data to the server. Over time, the server will assemble a fingerprint of the data across different times of the day. Using either SVM or logistic regression, the server can make prediction on stress. The results show that the precision can reach 68% if appropriate models are used.

Kwapisz et al. [75] proposed a user activity recognition mechanism based on phone accelerometers. The system first collects data from users who carried cell phone while performing some chosen activities. With the time series generated by the accelerometers, the system then transform them into information features, such as average, standard deviation, etc. At last, the system uses machine learning method, including logistic regression and multilayer perceptron, to classify the feature vectors into different activities.

The authors test the system with data collected from twenty-nine users. The results show that the precision can be over 90%, and the precision of multilayer perceptron based classification is higher than logistic regression based classification.

7.2 Industrial applications

Patil et al. [76] proposed an agriculture system that can monitor the environment conditions of vineyard, and predict the grape diseases in its early stages. The system used varieties of sensors to monitor the temperature, humidity and moisture throughout the yards. Using ZigBee, the data will be transmitted to servers, where a hidden markov model will be applied. In the hidden markov model, each state represents a certain condition. The author had implemented the system in real-world since Nov, 2015. The results show that the accuracy of the hidden markov model is 90.9%, which greatly improve the accuracy of statistical methods.

Siryani et al. [77] used machine learning to improve the efficiency of smart meter operation. With the tremendous increasing number of smart meters, administrators need to guarantee the cost efficiency of their operations. In this paper, the authors used varieties of machine learning methods, to predict whether to send a technician to a customer location. With higher accuracy, the system can reduce much travel expense and human resources.

The models were tested using data from a commercial network. Different classification algorithms, including bayesian network, naive bayes, decision tree and random forest, were tested. Finally, the results show that random forest achieve the highest accuracy, which is 96.69%, and the expected saving cost is about 1 million US dollars for the commercial network.

Ling et al. [78] designed an IoT-based system, that can detect occupancy of parking spaces automatically. The system first uploads the collected images. Then a vehicle recognition function is used to learn the parking spot. After that, a feature clustering algorithm based on Mean-shift, is used to find the most frequently parked locations.

The authors test the system on a Raspberry Pi 3 model. Camera data are collected on a local street near university of Washington campus. The Raspberry Pi board is connected to AWS IoT for restoring and observing. The results show that the real-time accuracy can achieve 97%.

Guo et al. [79] proposed an innovative method for detecting the characterization of flowering dynamics of rice. The method first collect time series of images from the rice fields. Secondly, the method extract local feature points from the images. During the third step, the method will generate visual words as the object-recognition approach. The method will use SVM to classify the time series of images, and detect the flowering part. For evaluation, the authors collected image data during different time with different rice varieties. The results show that the method perform well for counting number of flowering panicles. The accuracy of classification can be over 80% when proper training data are chosen.

7.3 Challenges and open issues

Saving computing resources Most IoT devices are equipped with lower battery and micro-controllers, even the gateway is limited by battery and computing resources. However, some machine learning method, e.g., DNN, needs lots of computing resources and is power hungry. Thus, how to distribute the tasks among different computing nodes, to save power and computing resources and achieve near-optimal accuracy, is an important research direction for IoT network applications.

Unstructured data sources Most works use structured data sources, such as sensors, images and records. However, in real IoT world, more data exist with unstructured format. It is worth further investigation on how to use machine learning based on these data.

Real-time and online analysis In many IoT applications, like health and industrial monitoring, the devices need to compute on-line and give feedback in real-time. Thus, the requirement for better security, QoS and computation complexity is higher than other applications. These problems need to be further addressed in the future.

8 Conclusion

Machine learning has a great potential to be the key technology for IoT. Machine learning trends to provide analytics for the IoT applications. Despite the recent wave of success of machine learning for networking, there is a scarcity of machine learning literatures about its applications for IoT services and systems, which this survey aims to address. This paper is different from the previously published survey papers in terms of focus, scope, and breadth; we have written this paper to emphasize the application of machine learning for IoT and the coverage of recent advances. Due to the versatility and evolving nature of IoT, it is impossible to cover each and every application. However, this paper has made an attempt to cover the major applications of machine learning for IoT and the relevant techniques, including traffic profiling, IoT device identification, security, edge computing infrastructure, network management based on SDN, and typical IoT applications. We have presented a thorough study on the recent researches about the application of machine learning for IoT, its technical progress, and application domains. We have also presented concise research challenges and open issues, which are critical to the application of machine learning for IoT.

Acknowledgements This work is partly supported by the National Natural Science Foundation of China(Grant Nos. 61772345, 61402294, 61672358 and 61502314), the Major Fundamental Research Project in the Science and Technology Plan of Shenzhen(Grant Nos. JCYJ20150324140036842, JCYJ20160310095523765, JCYJ20160307111232895 and JCYJ20160307115030281), the Science and Technology Research Project of Chongqing Municipal Education Commission of China(The Research on Data Integrity Detection based on the Cloud Storage, No. KJ1601401), and a grant from Innovation and Technology Fund of Hong Kong (Project No. ITS/304/16).

References

1. IoT Analytics. Why the internet of things is called internet of things: Definition, history, disambiguation. https://iot-analytics.com/internet-of-things-definition/
2. Gartner Research. Gartner says 6.4 billion connected things will be in use in 2016, up 30 percent from 2015. http://www.gartner.com/newsroom/id/3165317

3. Juniper Research. Internet of things connected devices to almost triple to over 38 billion units by 2020. http://www.juniperresearch.com/press/press-releases/iot-connected-devices-to-triple-to-38-bn-by-2020
4. The Statistics Portal. Internet of things (iot): number of connected devices worldwide from 2012 to 2020 (in billions). http://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/
5. Sharma SK, Wang X (2017) Live data analytics with collaborative edge and cloud processing in wireless iot networks. IEEE Access 5:4621–4635
6. Chau DH, Kittur A, Hong JI, Faloutsos C (2011) Apolo: Making sense of large network data by combining rich user interaction and machine learning. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI '11, Vancouver, BC, pp 167–176
7. Klaine PV, Imran MA, Onireti O, Souza RD (2017) A survey of machine learning techniques applied to self-organizing cellular networks. IEEE Commun Surv Tutor 19(4):2392–2431 (**Fourthquarter**)
8. Suthaharan S (2014) Big data classification: problems and challenges in network intrusion prediction with machine learning. SIGMETRICS Perform Eval Rev 41(4):70–73
9. Usama M, Qadir J, Raza A, Arif H, Yau KA, Elkhatib Y, Hussain A, Al-Fuqaha AI (2017) Unsupervised machine learning for networking: techniques, applications and research challenges. CoRR. arXiv:1709.06599 (**Online**)
10. Ayoubi S, Limam N, Salahuddin MA, Shahriar N, Boutaba R, Estrada-Solano F, Caicedo OM (2018) Machine learning for cognitive network management. IEEE Commun Mag 56(1):158–165
11. Fadlullah ZM, Tang F, Mao B, Kato N, Akashi O, Inoue T, Mizutani K (2017) State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE Commun Surv Tutor 19(4):2432–2455 (**Fourthquarter**)
12. Wang M, Cui Y, Wang X, Xiao S, Jiang J (2017) Machine learning for networking: Workflow, advances and opportunities. IEEE Netw 32(2):92–99
13. Hammerschmidt CA, Garcia S, Verwer S, State R (2017) Reliable machine learning for networking: key issues and approaches. In: 2017 IEEE 42nd conference on local computer networks (LCN), Singapore, pp 167–170
14. Casas P, Vanerio J, Fukuda K (2017) Gml learning, a generic machine learning model for network measurements analysis. In: 2017 13th international conference on network and service management (CNSM), Tokyo, pp 1–9
15. Claffy KC, Braun H-W, Polyzos GC (1995) A parameterizable methodology for internet traffic flow profiling. IEEE J Sel Areas Commun 13(8):1481–1494
16. Krishnamurthy B, Sen S, Zhang Y, Chen Y (2003) Sketch-based change detection: methods, evaluation, and applications. In: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement. ACM, New York, pp 234–247
17. 1. Lakhina A, Crovella M, Diot C (2004) Diagnosing network-wide traffic anomalies. In: ACM SIGCOMM Computer Communication Review, vol 34, no 4. ACM, New York, pp 219–230
18. Lakhina A, Crovella M, Diot C (2004) Characterization of network-wide anomalies in traffic flows. In: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement. ACM, New York, pp 201–206
19. Xu K, Zhang Z-L, Bhattacharyya S (2008) Internet traffic behavior profiling for network security monitoring. IEEE ACM Trans Netw 16(6):1241–1252
20. Hu Y, Chiu D-M, Lui JC (2009) Profiling and identification of p2p traffic. Comput Netw 53(6):849–863
21. Iliofotou M, Gallagher B, Eliassi-Rad T, Xie G, Faloutsos M (2010) Profiling-by-association: a resilient traffic profiling solution for the internet backbone. In: Proceedings of the 6th International Conference. ACM, New York, p 2
22. Iliofotou M, Kim H-C, Faloutsos M, Mitzenmacher M, Pappu P, Varghese G (2011) Graption: A graph-based p2p traffic classification framework for the internet backbone. Comput Netw 55(8):1909–1920
23. Brauckhoff D, Dimitropoulos X, Wagner A, Salamatian K (2012) Anomaly extraction in backbone networks using association rules. IEEE ACM Trans Netw (TON) 20(6):1788–1799
24. Huang N-F, Jai G-Y, Chao H-C, Tzang Y-J, Chang H-Y (2013) Application traffic classification at the early stage by characterizing application rounds. Inf Sci 232:130–142
25. Glatz E, Mavromatidis S, Ager B, Dimitropoulos X (2014) Visualizing big network traffic data using frequent pattern mining and hypergraphs. Computing 96(1):27–38
26. Bakhshi T, Ghita B (2015) User traffic profiling in a software defined networking context. In: International conference on internet technologies and applications, Wrexham, UK, September 8–11, pp 91–97. https://doi.org/10.1109/ITechA.2015.7317376
27. Kirchler M, Herrmann D, Lindemann J, Kloft M (2016) Tracked without a trace: linking sessions of users by unsupervised learning of patterns in their dns traffic. In: Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security. ACM, New York, pp 23–34
28. Das AK, Pathak PH, Chuah C-N, Mohapatra P (2017) Privacy-aware contextual localization using network traffic analysis. Comput Netw 118:24–36
29. Stolfo SJ, Hershkop S, Wang K, Nimeskern O, Hu CW (2003) Behavior profiling of email. In: Chen H, Miranda R, Zeng DD, Demchak C, Schroeder J, Madhusudan T (eds) Intelligence and security informatics, ISI 2003, Lecture notes in computer science, vol 2665. Springer, Berlin, Heidelberg, pp 960–960
30. Stöber T, Frank M, Schmitt J, Martinovic I (2013) Who do you sync you are?: smartphone fingerprinting via application behaviour. In: Proceedings of the sixth ACM conference on security and privacy in wireless and mobile networks. ACM, Budapest, Hungary, pp 7–12
31. Das A, Borisov N, Caesar M (2014) Do you hear what i hear?: fingerprinting smart devices through embedded acoustic components. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. ACM, Scottsdale, Arizona, pp 441–452
32. Bojinov H, Michalevsky Y, Nakibly G, Boneh D (2014) Mobile device identification via sensor fingerprinting. arXiv:1408.1416
33. Patel HJ, Temple MA, Baldwin RO (2015) Improving zigbee device network authentication using ensemble decision tree classifiers with radio frequency distinct native attribute fingerprinting. IEEE Trans Reliab 64(1):221–233
34. Huynh M, Nguyen P, Gruteser M, Vu T (2015) Poster: Mobile device identification by leveraging built-in capacitive signature. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. ACM, Denver, Colorado, pp 1635–1637
35. Tuama A, Comby F, Chaumont M (2016) Camera model identification based machine learning approach with high order statistics features. In: 2016 24th European signal processing conference (EUSIPCO). IEEE, Budapest, pp 1183–1187
36. Kurtz A, Gascon H, Becker T, Rieck K, Freiling F (2016) Fingerprinting mobile devices using personalized configurations. Proc Priv Enhanc Technol 2016(1):4–19
37. Baldini G, Dimc F, Kamnik R, Steri G, Giuliani R, Gentile C (2017) Identification of mobile phones using the built-in magnetometers stimulated by motion patterns. Sensors 17(4):783

38. Miettinen M, Marchal S, Hafeez I, Asokan N, Sadeghi A-R, Tarkoma S (2017) Iot sentinel: automated device-type identification for security enforcement in IoT. In: 2017 IEEE 37th international conference on distributed computing systems (ICDCS). IEEE, Atlanta, GA, pp 2177–2184
39. Meidan Y, Bohadana M, Shabtai A, Guarnizo JD, Ochoa M, Tippenhauer NO, Elovici Y (2017) Profiliot: a machine learning approach for IoT device identification based on network traffic analysis. In: Proceedings of the symposium on applied computing. ACM, Marrakech, Morocco, pp 506–509
40. Kotenko I, Saenko I, Skorik F, Bushuev S (2015) Neural network approach to forecast the state of the internet of things elements. In: 2015 XVIII international conference on soft computing and measurements (SCM), May 2015, St. Petersburg, pp 133–135
41. Baldini G, Giuliani R, Steri G, Neisse R (2017) Physical layer authentication of internet of things wireless devices through permutation and dispersion entropy. In: 2017 global internet of things summit (GIoTS), Geneva, pp 1–6
42. Sharaf-Dabbagh Y, Saad W (2017) Demo abstract: cyber-physical fingerprinting for internet of things authentication. In: 2017 IEEE/ACM second international conference on internet-of-things design and implementation (IoTDI), Pittsburgh, PA, pp 301–302
43. Jeong HJ, Lee HJ, Moon SM (2017) Work-in-progress: cloud-based machine learning for iot devices with better privacy. In: 2017 international conference on embedded software (EMSOFT), Seoul, pp 1–2
44. Jincy VJ, Sundararajan S (2015) Classification mechanism for iot devices towards creating a security framework. In: Buyya R, Thampi SM (eds) Intelligent distributed computing. Springer International Publishing, Cham, pp 265–277
45. Nobakht M, Sivaraman V, Boreli R (2016) 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), Salzburg, pp 147–156
46. Caedo J, Skjellum A (2016) Using machine learning to secure IoT systems. In: 2016 14th annual conference on privacy, security and trust (PST), Auckland, pp 219–222
47. Do VT, Engelstad P, Feng B, van Do T (2016) Strengthening mobile network security using machine learning. In: Younas M, Awan I, Kryvinska N, Strauss C, Thanh DV (eds) Mobile web and intelligent information systems. Springer International Publishing, Cham, pp 173–183
48. Stroeh K, Mauro Madeira ER, Goldenstein SK (2013) An approach to the correlation of security events based on machine learning techniques. J Internet Serv Appl 4(1):7
49. Rathore H, Jha S (2013) Bio-inspired machine learning based wireless sensor network security. In: 2013 world congress on nature and biologically inspired computing. IEEE, Fargo, ND, pp 140–146
50. Davis A, Parikh J, Weihl WE (2004) Edge computing: extending enterprise applications to the edge of the internet. In: International conference on World Wide Web—Alternate track papers & posters, WWW 2004. ACM, New York, NY, pp 180–187
51. Grewe D, Wagner M, Arumaithurai M, Psaras I, Kutscher D (2017) Information-centric mobile edge computing for connected vehicle environments: challenges and research directions. In: The workshop on mobile edge communications. ACM, Los Angeles, CA, pp 7–12
52. Borthakur D, Dubey H, Constant N, Mahler L, Mankodiya K (2017) Smart fog: fog computing framework for unsupervised clustering analytics in wearable internet of things. In: 2017 IEEE global conference on signal and information processing (GlobalSIP). IEEE, Montreal, QC, pp 472–476
53. Drolia U, Guo K, Narasimhan P (2017) Precog: prefetching for image recognition applications at the edge. In: ACM/IEEE symposium on edge computing. ACM, San Jose, California, pp 1–13
54. Azimi I, Anzanpour A, Rahmani AM, Pahikkala T, Levorato M, Liljeberg P, Dutt N (2107) HiCH: Hierarchical fog-assisted computing architecture for healthcare IoT. ACM Trans Embed Comput Syst 16(5s):174
55. Grassi G, Sammarco M, Bahl P, Jamieson K, Pau G (2015) Poster: Parkmaster: Leveraging edge computing in visual analytics. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, ser. MobiCom '15. ACM, New York, pp 257–259. https://doi.org/10.1145/2789168.2795174 (**Online**)
56. Wang S, Zhao Y, Huang L, Xu J, Hsu CH (2017) Qos prediction for service recommendations in mobile edge computing. J Parallel Distrib Comput. https://doi.org/10.1016/j.jpdc.2017.09.014
57. Zissis D (2017) Intelligent security on the edge of the cloud. In: International conference on engineering, technology and innovation. IEEE, Funchal, pp 1066–1070
58. Schneible J, Lu A (2017) Anomaly detection on the edge. In: MILCOM 2017—2017 IEEE Military Communications Conference (MILCOM), pp 678–682
59. Abeshu A, Chilamkurti N (2018) Deep learning: the frontier for distributed attack detection in fog-to-things computing. IEEE Commun Mag 56(2):169–175
60. Yang M, Zhu T, Liu B, Xiang Y, Zhou W (2018) Machine learning differential privacy with multifunctional aggregation in a fog computing architecture. IEEE Access 6:17119–17129. https://doi.org/10.1109/ACCESS.2018.2817523
61. Hogan M, Esposito F (2017) Stochastic delay forecasts for edge traffic engineering via bayesian networks. In: IEEE international symposium on network computing and applications. IEEE, Cambridge, MA, pp 1–4
62. Kim H, Feamster N (2013) Improving network management with software defined networking. Commun Mag IEEE 51(2):114–119
63. Kim HJ, Jung MY, Chin WS, Jang JW (2017) Identifying service contexts for qos support in iot service oriented software defined networks. In: Bouzefrane S, Banerjee S, Sailhan F, Boumerdassi S, Renault E (eds) Mobile, secure, and programmable networking. MSPN 2017, Lecture notes in computer science, vol 10566. Springer, Cham, pp 99–108
64. Vukobratovic D, Jakovetic D, Skachek V, Bajovic D, Sejdinovic D, Kurt GK, Hollanti C, Fischer I (2016) Condense: a reconfigurable knowledge acquisition architecture for future 5g iot. IEEE Access 4:3360–3378
65. Jagadeesan LJ, Mendiratta V (2016) Programming the network: application software faults in software-defined networks. In: 2016 IEEE international symposium on software reliability engineering workshops (ISSREW). IEEE, Ottawa, ON, pp 125–131
66. Taneja M (2016) A framework for traffic management in iot networks. In: 2016 2nd international conference on contemporary computing and informatics (IC3I). IEEE, Noida, pp 316–323
67. Uwagbole SO, Buchanan WJ, Fan L (2017) An applied pattern-driven corpus to predictive analytics in mitigating sql injection attack. In: 2017 seventh international conference on emerging security technologies (EST). IEEE, Canterbury, pp 12–17
68. Ahmed ME, Kim H, Park M (2017) Mitigating dns query-based ddos attacks with machine learning on software-defined networking. In: MILCOM 2017—2017 IEEE military communications conference (MILCOM). IEEE, Baltimore, MD, pp 11–16
69. Bhunia SS, Gurusamy M (2017) Dynamic attack detection and mitigation in iot using sdn. In: 2017 27th international telecommunication networks and applications conference (ITNAC). IEEE, Melbourne, VIC, pp 1–6
70. Asthana S, Megahed A, Strong R (2017) A recommendation system for proactive health monitoring using IoT and wearable technologies. In: 2017 IEEE international conference on AI mobile services (AIMS). IEEE, Honolulu, HI, pp 14–21

71. Walinjkar A, Woods J (2017) ECG classification and prognostic approach towards personalized healthcare. In: 2017 international conference on social media, wearable and web analytics (Social Media). IEEE, London, pp 1–8
72. Nguyen HH, Mirza F, Naeem MA, Nguyen M (2017) A review on iot healthcare monitoring applications and a vision for transforming sensor data into real-time clinical feedback. In: 2017 IEEE 21st international conference on computer supported cooperative work in design (CSCWD). IEEE, Wellington, pp 257–262
73. Madeira R, Nunes L (2016) A machine learning approach for indirect human presence detection using IoT devices. In: 2016 eleventh international conference on digital information management (ICDIM). IEEE, Porto, pp 145–150
74. Pandey PS (2017) Machine learning and iot for prediction and detection of stress. In: 2017 17th international conference on computational science and its applications (ICCSA). IEEE, Trieste, pp 1–5
75. Kwapisz JR, Weiss GM, Moore SA (2011) Activity recognition using cell phone accelerometers. SIGKDD Explor Newsl 12(2):74–82
76. Patil SS, Thorat SA (2016) Early detection of grapes diseases using machine learning and IoT. In: 2016 second international conference on cognitive computing and information processing (CCIP). IEEE, Mysore, pp 1–5
77. Siryani J, Tanju B, Eveleigh TJ (2017) A machine learning decision support system improves the internet of things smart meter operations. IEEE Internet Things J 4(4):1056–1066
78. Ling X, Sheng J, Baiocchi O, Liu X, Tolentino ME (2017) Identifying parking spaces detecting occupancy using vision-based IoT devices. In: 2017 global internet of things summit (GIoTS). IEEE, Geneva, pp 1–6
79. Guo W, Fukatsu T, Ninomiya S (2015) Automated characterization of flowering dynamics in rice using field-acquired time-series rgb images. Plant Methods 11(1):7

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.