IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 12, 2021, accepted October 24, 2021, date of publication November 9, 2021,
date of current version November 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3126834

Machine Learning in Network Anomaly Detection: A Survey

SONG WANG[1], (Member, IEEE),
JUAN FERNANDO BALAREZO[1], (Graduate Student Member, IEEE),
SITHAMPARANATHAN KANDEEPAN[1], (Senior Member, IEEE),
AKRAM AL-HOURANI[1], (Senior Member, IEEE),
KARINA GOMEZ CHAVEZ[1], AND BENJAMIN RUBINSTEIN[2], (Member, IEEE)
[1]School of Engineering, RMIT University, Melbourne, VIC 3000, Australia
[2]School of Computing and Information Systems, The University of Melbourne, Melbourne, VIC 3010, Australia

Corresponding author: Song Wang (s3478896@student.rmit.edu.au)

The work of Song Wang and Juan Fernando Balarezo was supported by the Australian Government Research Training Program Scholarship.

ABSTRACT Anomalies could be the threats to the network that have ever/never happened. To protect networks against malicious access is always challenging even though it has been studied for a long time. Due to the evolution of network in both new technologies and fast growth of connected devices, network attacks are getting versatile as well. Comparing to the traditional detection approaches, machine learning is a novel and flexible method to detect intrusions in the network, it is applicable to any network structure. In this paper, we introduce the challenges of anomaly detection in the traditional network, as well as in the next generation network, and review the implementation of machine learning in the anomaly detection under different network contexts. The procedure of each machine learning category is explained, as well as the methodologies and advantages are presented. The comparison of using different machine learning models is also summarised.

INDEX TERMS Machine learning, anomaly detection, network security, software defined network, Internet of Things, cloud network.

I. INTRODUCTION

Network security has become increasingly critical these days, from the traditional computer network and cellular network to the next generation software defined network (SDN) and Internet of Things (IoT). The rapid growing network brings efficiency and convenience to our life, as well as the demand for high quality of service. Even though the network use case is getting more complex and a network device needs to process more data, users hope to get responses more quickly and show a lower tolerance to the service interruption. Firewalls, deep packet inspection (DPI) systems and intrusion detection systems (IDS) are the typical methods for anomaly detection, however, the cost to deploy these countermeasures and the complexity of system have to be considered [1], [2]. The security issue arises along with the evolution of network, the diversity of network services and applications provides hackers more opportunities to compromise the network than ever before. Especially, the working procedure in the next generation network, is quite different from the legacy network, current anomaly detection methods need upgrade to adapt to the change in these networks. For example, SDN decouples the control plane from the data plane, a centralised controller is usually responsible for the management of multiple data plane devices, besides a higher work load comparing to the control plane of a legacy router/switch, this architecture brings new challenges that the entire network is impacted if the controller is compromised. And the interaction between the control and data plane is no longer within the same hardware, it is mostly going through a network so that the administrator has to consider the security of data transmission, as a command from the controller towards the forwarding device could be modified during transfer. Similarly, data storage in the cloud network is quite different from the past, data has to pass through the network before being stored in a remote server. Also, IoT aims to connect everything from

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

everywhere, the diversity of IoT applications increases the number of devices in the network, as well as complicates the network architecture. Due to the large number of connected devices and the high-speed broadband, anomaly detection requires to process big data over a complex network structure with a prompt reaction. This has become one of the biggest challenges to protect networks [3]–[5].

Machine learning (ML), as an analytical tool based on statistics, has been widely discussed and deployed in various areas. Its capability to make decisions after study and analysis relieves people from processing a huge amount of data, so that ML is normally used to investigate complicated scenarios. Furthermore, its response to abnormal behaviours is usually much quicker than human beings, which is an advantage in early detection. For known attacks, ML gains experience from existing records to understand their characteristics; while for unknown attacks, ML finds the outlier from the intrinsic patterns of data. ML can create diverse models with various algorithms, the way to work with these models also has a big difference. Based on the available dataset, the network operator could choose supervised learning to train a predictor when the size of labelled data is large, or a semi-supervised learning model when the number of labelled data is limited. Even if running the same model to detect the same type of attack, the outcome varies depending on the features that you prefer ML to consider [6], [7]. As a matter of fact, the most difficult step using ML is data preparation, from data collection to annotation, a high quality dataset is vital to the prediction. Because the output of ML highly relies on the data from which algorithms learn the skill to distinguish normal operations from anomalous behaviours. Thus, in this paper, we introduce ML algorithms, as well as discuss the implementation of ML models in anomaly detection under different network contexts.

The contributions of this article are:

- It presents a comprehensive survey on the ML types.
- Detailed review and discussion of ML techniques in anomaly detection are introduced.
- Various network scenarios employing ML for anomaly detection are analysed.
- Characteristics and advantages of each ML model in anomaly detection are summarised.

The rest of this article is organised as follows. Related works are presented in Section II. Section III introduces four ML types and their procedures in anomaly detection. Then Section IV reviews the challenges of anomaly detection under various network contexts. Detailed survey and comparison of existing solutions are in the Section V. Finally, Section VI concludes this article.

II. RELATED WORK

ML has been applied to security in various types of networks, from the traditional computer network to the IoT network, and there have been surveys discussing the existing solutions, some recent review papers from major journals are listed below. Buczak *et al.* [8] focused on the intrusion detection system using supervised and unsupervised learning in the cyber network. Hodo *et al.* [9] reviewed the ML techniques in the IDS under computer, cloud and IoT networks, the feature selection when training a supervised learning model or classifying traffic in a unsupervised learning model is also discussed to show its importance in the ML based IDS. Da Costa *et al.* [10] surveyed the intrusion detection using ML applications under the context of IoT. Ucci *et al.* [11] and Gibert *et al.* [14] researched malware detection and classification in the Windows system with supervised and unsupervised learning. They talked about the features a malware is interested in, and how the ML algorithms are used to classify a malware. Tahsien *et al.* [12] and Hussain *et al.* [13] described the potential threats per IoT layer and introduced the principle of some ML algorithms, then ML applications are presented to solve these issues. Nassif *et al.* [15] studied the threats in the cloud network, and the way to secure cloud network using supervised learning.

Although ML techniques have been deployed in diverse domains for addressing security issues, most of the surveys only focused on a specific network type, there is no comprehensive study on the anomaly detection using four types of ML models under different network environment. Hence, in this paper, we describe how supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning can be used in the anomaly detection, and the advantages and disadvantages of existing solutions in the computer network, cellular network, SDN, IoT and cloud network. Moreover, we also include some detection methods which are proposed in general without specifying any network type, these solutions are validated by experimental dataset. The comparison of our paper and existing survey papers are summarised in Table 1. And the taxonomy of this survey is illustrated in Figure 1.

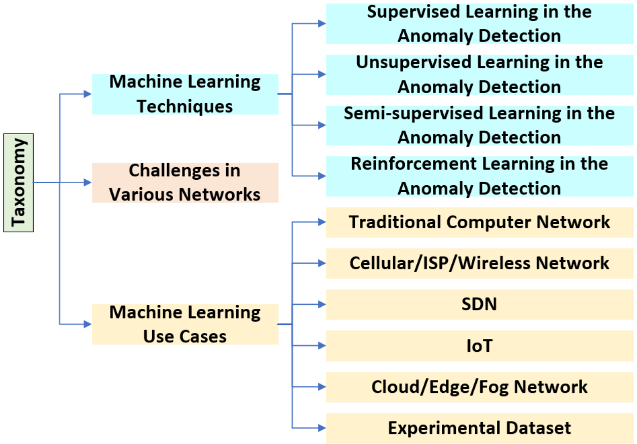

Taxonomy
Machine Learning Techniques
Supervised Learning in the Anomaly Detection
Unsupervised Learning in the Anomaly Detection
Semi-supervised Learning in the Anomaly Detection
Reinforcement Learning in the Anomaly Detection
Challenges in Various Networks
Machine Learning Use Cases
Traditional Computer Network
Cellular/ISP/Wireless Network
SDN
IoT
Cloud/Edge/Fog Network
Experimental Dataset

FIGURE 1. Taxonomy of this survey paper.

III. BACKGROUND OF MACHINE LEARNING (ML) TECHNIQUES

As a subset of Artificial Intelligence (AI), ML is a powerful tool that can be used for network anomaly detection via

TABLE 1. Comparison of this paper with other survey papers.

Papers	Published In	Year	ML Techniques	Focused Areas
Buczak *et al.* [8]	IEEE COMST	2016	Supervised and unsupervised learning	Cyber analytics for intrusion detection
Hodo *et al.* [9]	ArXiv	2017	Supervised and unsupervised learning	Intrusion detection in the cloud, IoT and computer network
Da Costa *et al.* [10]	Elsevier Computer Networks	2019	Supervised, unsupervised and semi-supervised learning	Intrusion detection in the IoT network
Ucci *et al.* [11]	Elsevier Computers and Security	2019	Supervised, unsupervised and semi-supervised learning	Malware analysis in Windows
Tahsien *et al.* [12]	Elsevier Network and Computer Applications	2020	Supervised, unsupervised and reinforcement learning	IoT security
Hussain *et al.* [13]	IEEE COMST	2020	Supervised, unsupervised, semi-supervised and reinforcement learning	IoT security
Gibert *et al.* [14]	Elsevier Network and Computer Applications	2020	Supervised and unsupervised learning	Malware detection and classification in Windows
Nassif *et al.* [15]	IEEE Access	2021	Supervised learning	Cloud security
This paper	-	2021	Supervised, unsupervised, semi-supervised and reinforcement learning	Anomaly detection in cellular networks, SDN, IoT networks, cloud networks and traditional computer networks

scientific study of traffic samples, this procedure is quite different under each ML category. Even running the same ML model with two identical datasets, the performance may vary from the way a ML algorithm is used, such as the features chosen from the dataset or the weight defined for each feature. More features do not always mean better results, instead it could lead to overfitting in the model [16]. Thus, it is worth reviewing and comparing current solutions so as to better understand and build a model with available ML techniques and data in hand. ML can be classified into four categories as shown in Figure 2: (i) supervised learning (SL), (ii) unsupervised learning (UL), (iii) semi-supervised learning (SSL), and (iv) reinforcement learning (RL).

(i) **Supervised Learning (SL)** learns from existing labelled datasets, which is called training set, and by comparing with the known labels the predicted output can be evaluated. Past experience is used as a reference to make a decision, and a high quality training set is always essential to build a well-performed model, however, a satisfying result is not guaranteed by the dataset only, the training method is another key factor in building a trustworthy predictor. In the SL, a classifier model is created through training first, after that it is able to predict either discrete or continuous outputs. Before prediction, the performance, such as accuracy, of a SL model is usually validated to show its reliability. SL can also be divided into classification and regression techniques [17].

The classification technique classifies input data into discrete categories, it calculates the probability of a test sample to be under each category, and the one with most votes wins [18]. This probability is the likelihood of a sample belonging to a class. Typical applications including medical imaging and credit scoring. The regression technique predicts continuous responses, usually quantities, from the input variables, for example, changes in temperature or fluctuations in power demand [19]. Typical applications include electricity load forecasting and algorithmic trading. To evaluate these two techniques, the classification model can check the percentage of correct predictions; while the regression model could calculate the root-mean-square error, because the output is continuous, a deviation between the prediction and real value is acceptable.

(ii) **Unsupervised Learning (UL)** finds hidden patterns or intrinsic structures in data to group them, it has input data but no expected output variables. Unlike SL, there is neither labelled sample nor training process, which is to say it works on its own and its performance can hardly be evaluated. Although some researchers use existing labelled data in the UL model to verify its outcome, this is unable to achieve in the real implementation, and sometimes experts have to analyse the result manually to run an external evaluation. UL is mainly used for clustering and dimensionality reduction. In the cluster problem, it uses clustering techniques so that one sample may belong to one cluster only or multiple clusters; while in the dimensionality reduction, UL identifies the correlated features in the dataset, so that redundant information can be removed to reduce the noise. Typical applications include market research, and object recognition [20].

(iii) **Semi-supervised Learning (SSL)** combines both labelled and unlabelled data to build the classifier, which is suitable for the scenario that has a paucity of labelled dataset. It employs the training process as mentioned in the supervised learning to prepare a predictor with limited labelled data, and this predictor classifies unlabelled samples, then each pseudo labelled sample is given a confidence value to tell the administrator whether this prediction is assured. Those confident samples will

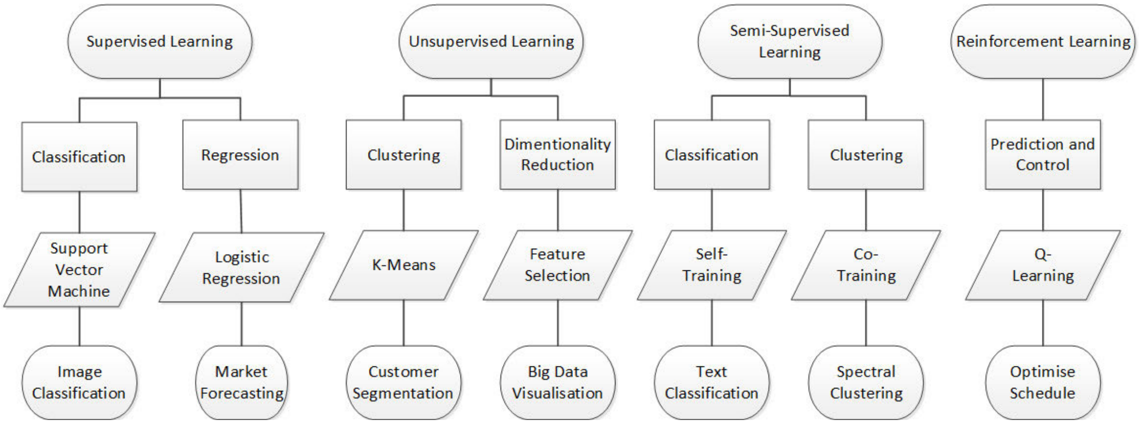


FIGURE 2. Machine Learning (ML) categories, typical algorithms and use cases.

join the new training set to update the classifier until all the data have labels. As unlabelled data is actually tagged randomly in the prediction, assumptions, such as smoothness and cluster, have to be made prior to the training of unlabelled instances [21], [22].

(iv) **Reinforcement Learning (RL)** uses states, actions, and rewards to judge if the machine has made a good decision. The algorithm used in RL is called an agent, and the agent is working in the object, called environment. At first, the environment sends the current state to the agent, and the agent chooses actions in response to that state, so that it enters a new state based on the action. Then, the environment sends this new state and a reward to the agent. This loop keeps running until the agent receives a terminal state. Through the rewards given by the environment, the agent develops an optimal policy to achieve the maximum long-term rewards [23].

To evaluate the performance of ML models, there are four main metrics: accuracy, precision, recall and F-measure. The procedure of running ML in anomaly detection is summarised as follows.

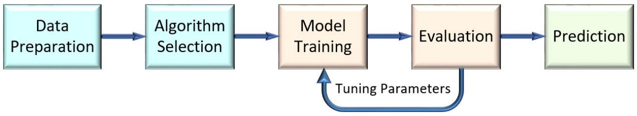


FIGURE 3. Supervised learning procedure.

A. SL IN THE ANOMALY DETECTION

In general, the process of anomaly detection using SL is shown in Figure 3, it includes: data preparation, algorithm selection, model training, evaluation, model improvement and prediction. Data preparation is the most important and time-consuming step, from data collection to annotation. The collected data is not ready to work in the SL, duplicated data shall be removed, features must be extracted and converted to the format that can be understood by the SL algorithm. Besides, a classifier is added to each sample to prepare a group of labelled data. This data group is further split into training set and validation set. Once the SL algorithm is chosen, a predictor is trained via the training set, and it is then evaluated through the validation set. Parameters of a SL algorithm can be adjusted to reach the best outcome according to the result of evaluation. In the end, the trained model is able to predict samples in real time [24].

Rather than using all the features in the dataset, select only the key features for training and prediction is a better option, because it filters features that are not strong related to the output, and facilitates an enhanced understanding of the model [25]. Sometimes the performance of prediction improves, and sometimes even though the outcome impairs the degradation of prediction is very limited. Additionally, this saves system resources and time in training. Techniques to extract features are described below:

1) Wrapper approach searches for essential features by evaluating the output using the predictor itself. The entire feature group is rearranged into several subsets, and the subset which has the lowest estimated error is considered as the most related features in the prediction [26]. Genetic algorithm (GA) [27] and recursive feature elimination [28] can be applied in the anomaly detection application.
2) Filter approach assesses feature importance via the characteristics of dataset, such as correlation, and the predictor is ignored in this method [29]. Typical algorithms include fisher score [30] and correlation based feature selection [31].
3) Embedded method is a trade off between the previous two methods, because the computational cost in wrapper method is high; while the selected features using filter method may not be optimal. Thus, embedded method picks features in filter mode and validate the performance in wrapper mode [32], [33]. Lasso is a typical algorithm that can be employed in anomaly detection [34].

Apart from the measures above, ensemble is also used to gain a more stable and reliable model, two typical ensemble types are bagging and boosting. Bagging method trains classifiers independently and votes with equal weight, it reduces variance in the model [35]; while boosting method trains a new model based on the previous model, it has low bias in the model [36].

B. UL IN THE ANOMALY DETECTION

As no training with labelled data is performed in the UL for anomaly detection, finding outliers in the data is based on the assumption that abnormal behaviours rarely occur. The procedure is given in Figure 4, similar to SL, data are collected and adapted to the form that can be understood by the UL algorithm, but no label is required. Comparing to SL, UL is able to process computationally complex cases, because it is data-driven and can handle unknown scenarios. On the basis of the feature of anomaly and the principle of UL algorithm, a specific attack is more likely to be detected by certain UL models, which is to say that selecting a suitable algorithm is also necessary for anomaly detection [37]. Moreover, feature extraction and normalisation are usually performed on the data before sending to the UL clustering models. Numerical data are always preferred in the test, such as IP address and number of bytes, because they are valuable information in a cluster [38], [39].

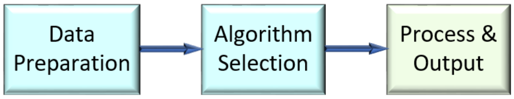


FIGURE 4. Unsupervised learning procedure.

In the real implementation, the accuracy of an UL clustering model is hard to evaluate, the outcome might be untrustworthy. However, with labelled data, the performance of UL in the anomaly detection is proved to be satisfying [40]. Especially in front of unknown attacks, UL has its advantages over SL. Since SL models rely on the training data, unknown attacks might slip through the net due to the lack of related records, and UL models could step in to detect the issue.

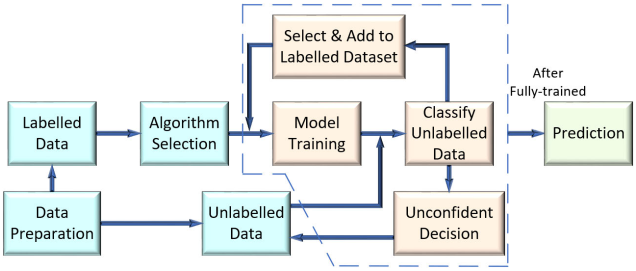


FIGURE 5. Semi-supervised learning procedure.

C. SSL IN THE ANOMALY DETECTION

The procedure of using SSL is illustrated in Figure 5, apart from the data preparation of labelled and unlabelled data, SSL trains a model with labelled data first. These labelled data could be in multiple classes, which means the training set has samples of all the attack types; or in one class, i.e. normal samples, which is to say the predictor is trained by normal traffic only and needs to classify anomalous traffic. As a trade-off between SL and UL, SSL is more applicable in the real world, because it is an option to obtain a relatively reliable prediction with a small number of data. SSL relieves the lack of labelled data and ensures the model has adequate training before implementation, however, incorrect classification of the unlabelled data could mislead the model to false prediction [41].

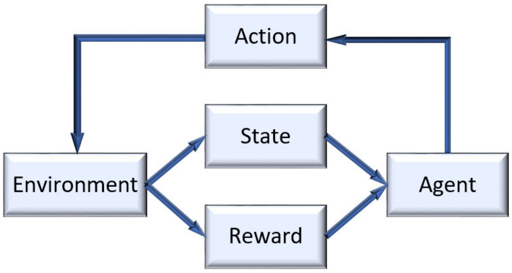


FIGURE 6. Reinforcement learning procedure.

D. RL IN THE ANOMALY DETECTION

RL is a mistake-driven learning method, which is depicted in Figure 6, this learning style is quite similar to the learning of human beings. One of the challenges to leverage RL in anomaly detection is the definition of RL parameters: state, action and reward. Although it does not require data labelling, there are too many features in the network that can become the state in RL, as well as the reward after each operation. These parameters determine the performance of RL in identifying malicious behaviours in the network.

Unlike other ML types, errors can be corrected in the RL, the machine "realises" its mistake from rewards or long-term returns, and then it will avoid these actions under the specific environment. Since RL is learning through the interaction with network for anomaly detection, a large amount of data and computing resources are required to achieve an ideal solution.

The difference of using the four ML types for anomaly detection in general is summarised in Table 2.

IV. ANOMALY DETECTION CHALLENGES IN VARIOUS NETWORKS

Before we explain ML for anomaly detection, challenges under different network contexts, including computer network, cellular network, SDN, IoT and cloud network, are discussed. Although cyber security has been researched for years, there are still open issues and challenges in different types of networks. In the traditional computer network, the intrusion detection system (IDS) is a typical countermeasure deployed to protect the network, especially in the large scale network, it is a mature system against threats. However, there are still some challenges in the IDS when protecting

TABLE 2. Comparison of supervised, unsupervised, semi-supervised and reinforcement learning in anomaly detection.

ML Types	Summary	Characteristics	Advantages	Disadvantages
SL	Predicting after trained by plenty of labelled dataset.	Performance can be validated via labelled test data.	The definition of attack types can be specific via the training set. More reliable when prediction is made in a similar situation to the training set.	False prediction rate increases with unfamiliar data. Labelled data is rare and the cost of labelling is high in the real world. Computational cost is high during training, especially with large data size. Unable to handle complex tasks.
UL	Categorise unlabelled data from features given.	Decision can be made without labelled data.	Able to detect novel threats if their features are quite different from the normal ones. Getting unlabelled data is much easier. Can handle complex scenarios. Quicker response in classification than SL.	Unable to know the performance due to the absence of labelled data. It may be a costly affair to analyse the output when identifying the threat type in a complicated scenario.
SSL	Initialise supervised learning with a small group of labelled data, and then classify unlabelled data accordingly.	Expand training set with high confident unlabelled data. The final model is trained by labelled and pseudo labelled data.	Obtain more confidence from labelled data than in the UL. Labelling limited size of data is acceptable in the real world.	Employment of incorrect predicted unlabelled data could mislead the classifier to make wrong decisions.
RL	Use trial and error method to try all the possible state-action pairs so as to find the strategy with a best long-term return.	Use the concept of reward to judge a response to the environment. Emphasise the final outcome rather than a single instant output.	Applicable to complicated real world problems that require the best results after a series of operations rather than a single action.	Resource consumption is high, because it is going to try all the state-action pairs.

the traditional network. The three key factors in evaluating an IDS are: accuracy, completeness and performance. The accuracy and completeness are hard to measure, and most of the evaluations are done by contrived dataset, which is hard to be unbiased and comprehensive. The complexity of IDS also increases in order to cover more attack scenarios. Furthermore, since new attacks are introduced and existing attacks are changing their methods, to update the profiles in the IDS after an unknown attack been detected, or to update the IDS itself to adapt to the change of attack method is not an easy task [2], [42].

Unlike legacy computer networks, devices in the cellular/wireless network are usually wireless connected and have mobility, a secure access authentication is essential to alleviate the probability of threats, such as DoS/DDoS attacks. Moreover, the large number of applications running in the cell phone provides a big opportunity for malware. Due to the diversity of services on the cell phone, the network is getting increasingly complex. Although the network can mostly work around the issue to maintain the normal connection, the anomaly detection still requires a high human workload. Because the analysis and localisation of the root cause are time-consuming, even for experienced engineers [43].

Both SDN and IoT networks have not been widely deployed yet, so that not much experience in these two networks. However, from the features and network architecture, some challenges can still be concluded. For SDN, the centralised control brings scalability issues, especially against flooding attacks [44]. The open source platform allows various detection methods to be implemented in the network, in the meanwhile, how to avoid the conflict between these methods is a potential challenge. For IoT, end devices are usually lack of security features due to energy efficiency, so the placement of anomaly detection system needs to be considered. Moreover, the detection range is also challenging, as existing solutions only target specific attacks, they need to be combined over the entire network [45].

IDS is also employed in the cloud network for anomaly detection, and since a cloud network consists of multiple components, the IDS needs to be configured in each component. Thus, the position of IDS is a challenge, and the work load of configuration is heavy [46].

In simple terms, the anomaly detection in the network is usually achieved through condition monitoring, the network state is defined by comparing the measurement with the maximum and minimum boundaries. To summarise, the key challenges in the existing anomaly detection solutions are the complexity of system and adaptability to the diversity of attacks. While a ML model is able to solve these issues, because the model can be updated and improved through learning, however, the performance still needs evaluation in each network.

V. ML FOR ANOMALY DETECTION USE CASES

In this section, we describe the ML applications in anomaly detection under different network domains.

A. TRADITIONAL COMPUTER NETWORK

Although countermeasures against attacks in the cyber network have been researched for many years, both these solutions and hackers are getting sophisticated, and the network scenario is becoming complex as well. ML involves more automatic quick responses to the change in the network, as well as in anomalous behaviours. Hamamoto *et al.* [47]

TABLE 3. Anomaly detection in the computer networks using ML.

Papers	ML Models	Anomaly	Methodology	Advantages	Disadvantages
Hamamoto *et al.* [47]	UL: GA and FL	Denial of service (DoS) and Distributed DoS (DDoS) attacks	Use GA to analyse network behaviour, and FL for prediction.	High performance in attack detection in an unsupervised manner.	Recall is only 76.5% which means the false negative rate is high.
Gu *et al.* [48]	SSL: K-Means	DDoS attacks	Cluster data and find the centre of each cluster, then classify data into a cluster referring to the distance to the cluster centre.	True positive rate in detection is high.	False positive rate is unstable, it varies from 0 to over 28%.
Alauthman *et al.* [49]	RL: NN	Botnet attacks	Use the output of a SL model as the state in the RL model, both the SL and RL model are improving through this interaction.	Good accuracy rate when the input data is reduced in the model, and training time is reduced.	Validated in the Matlab by three datasets, not implemented in the real network.
Smadi *et al.* [50]	RL: NN	Phishing emails	Use the reward from RL model to determine if current SL model is the best.	Can automatically adapt to any change in the network. The SL model is improved through the interaction with the RL model.	Require much more features than other works.
Xu *et al.* [51]	RL: Q-learning	General network anomalies	Use the RL model to adjust parameters in the anomaly detection models, and find the optimal model who has the highest reward.	The optimal configuration of an anomaly detection model is found through the RL model.	Recall and precision are unstable. The difference between the best and worst outputs could be around 20%.
Sethi *et al.* [52]	RL: Deep Q-network	Various attacks, such as DoS attacks	Run multiple agents under the same network to determine the network states, and reward the RL model from its prediction.	High accuracy in the detection.	The accuracy is low and false positive rate is high when tested using NSL-KDD and UNSW-NB15 datasets.
Jin *et al.* [53]	RL: Deep Deterministic Policy Gradient	Insider threats	Take flows, bandwidth and user reputation into consideration to train a RL model to optimise traffic scheduling policy.	Flexible and automatic security traffic management with fast response.	Validated in the simulator by a dataset, not implemented in the real network.

group network flows by time intervals, and extract key features, such as bits per second and source IP addresses. For numeric data, they can be used in the model directly; while for nominal values, entropy is calculated to represent the distribution of a specific value within the time interval. GA learns the behavioral pattern of traffic and predicts the network behavior. Based on the output from GA, fuzzy logic (FL) evaluates whether the traffic flow is abnormal in a time interval. Normally, labelled data and unlabelled data are processed separately in SSL, Gu *et al.* [48] cluster normal and abnormal data by the small amount of tagged samples using K-Means algorithm, the density of each data point within a specific radius is computed to find the cluster centre. With unlabelled data, the centre of each cluster is updated until convergence. To detect anomaly in the network, the distance between the data feature and each cluster centre is calculated, and the data is classified into the cluster with the shortest distance.

Sometimes the state or environment of RL model could be difficult to describe, so that the definition with the aid of other ML algorithms becomes a feasible solution. Alauthman *et al.* [49] use the output of NN to be the host state in a RL model for botnet detection, this state contains two sub-states which are the probability of being malicious and legitimate. The highest expected reward is then extracted depends on which probability is higher. A better NN policy will replace the old one, as well as the new behaviours that get a higher rewards will join the training set. Hence, the RL agent is improving to create a superior detector. Also, the RL model is used to enhance the detection performance in other models. Smadi *et al.* [50] train a NN model to outline the email filter system, and a RL model is employed during the training of NN model. For each training, neurons in the current NN model are updated, and a reward is given based on the output. A NN model with a higher reward always replaces the old model until the preset round of training is hit or the termination condition is met. Xu *et al.* [51] employ RL to adjust anomaly detection modules, which contain a variety of detection algorithms, to find the optimal strategy. Before adjustment, the implementation of a strategy is set as the target state, and the parameter adjustment is defined as an action in the RL model. Actions are taken iteratively until the state of anomaly detection module hits the target, or the predefined number of iteration is reached. With multiple attempts using various anomaly detection strategies, the one with largest accumulative rewards is the optimal strategy. To determine the reward in RL, Sethi *et al.* [52] adopt IDS to grade the output from RL agents. Multiple agents are deployed in routers under the same context of network, and each agent has several classifiers to predict the RL state. The state vector, which consists of the output from classifiers along with feature vector, is fed to the deep Q-network to obtain a Q-value. The action function makes decision on whether it is an attack or not through the comparison of Q-value and threshold. A positive or negative reward is received if the classification is the same as the actual result, which is given by the IDS.

Besides detecting abnormal behaviours, ML can also be used in the network management to avoid potential threats. Jin and Wang [53] employ RL to find the best scheduling policy to manage intranet traffic with the consideration of security. Each user has a reputation value to indicate how trustworthy his traffic is. The state in the RL is represented by the available bandwidth of links and the flows that are waiting to be scheduled. Actions are given per flow in the proposed model, and each action is comprised of the bandwidth allocation to this flow. The performance of scheduler is rewarded by the utilisation of links, length of queue, latency and the user trust level. This RL model considers security, performance and user requirement in the network when defending threats from inside. The ML applications in the traditional computer networks are summarised in Table 3.

B. CELLULAR/INTERNET SERVICE PROVIDER NETWORK/WIRELESS NETWORK

Comparing to the computer network, cellular/wireless networks are more wireless connection oriented. Because of the transmission medium, devices and links are more vulnerable to the attacks than using wired connection. And for cellular network, latency shall be much lower than in the computer networks because of voice services, while ML is able to provide a low detection delay approach. Malicious mobile applications usually generate benign traffic which far outweighs anomalous traffic, so that imbalanced data becomes a problem in the data analytic, because there is not enough information that can clearly indicate the abnormal behaviour. However, ML can overcome this challenge, Chen *et al.* [54] identify malicious behaviours in the cellular network so as to detect malicious applications. Based on the destination IP address and domain name in the packet, most SL algorithms have an excellent accuracy in judging malwares in the cell phone. Otoum *et al.* [55] deploy a RL model in the wireless sensor network to detect anomalies. The cluster head is elected based on the factors, such as the connectivity of a node and signal strength. Then the cluster head collects sensed data and redirects them to the RL model for analysis. The RL model makes decision on whether a sensor is behaving abnormally and the reward is given accordingly.

As supervised learning relies on labelled training set, on the one hand, its response to the anomalies might be slow due to the lack of abnormal samples. Hence, Dromard *et al.* [56] involve grid incremental clustering algorithms in the UL to rapidly detect any abnormal state in the network, so that real-time detection is achievable. The entire dataset is partitioned into cells, and each cell contains a subset of the original dataset. Then, dense cells who have a common face are grouped to form a cluster, which reduces the complexity comparing to handle the whole dataset. When new data come into the network, the update only happens in the previous feature space partition so that the computation is finished fast. On the other hand, annotating a large scale dataset is a big challenge in SL, so Al Mamun and Valimaki [57] propose an automatic labelling algorithm for applying anomaly detection in the cellular network. This algorithm considers two factors to classify a sample: range of KPI value and time series profile. A threshold is defined for KPI value to determine whether it is normal, while for time series profile the mean value and standard deviation are considered. Only when both of these two factors are abnormal, the sample is categorised as anomaly.

Without labelled data, UL is also a good option. Dey *et al.* [58] filter man-in-the-middle (MitM) attacks through profiles and features of incoming traffic. The operating system and coarse location of a client are utilised to determine whether the request is suspicious. And then an unsupervised clustering algorithm based on inter-packet delay further inspects the traffic. Hoang *et al.* [59] propose a simple method to detect eavesdropping attacks based on one-class labelled data, which only known as normal, using UL. An area that contains normal data is defined by one-class SVM (OCSVM) first. Then, unlabelled data are divided into two groups via K-Means model. For those data that sit within the predefined area, they are labelled as normal, or abnormal otherwise if outside the area.

Although UL can work solo to analyse problems, combining it with other ML techniques could result in a superior output [62]. Qu *et al.* [60] combine Mean Shift Clustering Algorithm (MSCA) and SVM to detect unknown attacks in the wireless sensor network, MSCA distinguishes attacks through abnormal features, and SVM is employed to maximise the margin between normal and attack features, so that the error in classification is minimised.

Ensemble method is another approach to improve predictive performance, a training set is divided into several small subsets, and one or multiple SL algorithms generate several classifiers via training by these subsets. The final prediction is given by combining the outputs from classifiers, i.e. the one with most votes wins [63], [64]. Vanerio and Casas [61] use a supervised learning model, called Super Learner, to enhance anomaly detection with ensemble learning approach. Super Learner is able to find the best combination of a group of basic prediction algorithms. Through the evaluation over a semi-synthetic dataset [65] which records traffic in the cellular network, results are better than using a single prediction model. Existing solutions are summarised in Table 4.

C. SDN

The programmability of SDN simplifies the implementation of ML than in other networks. ML security applications can be developed and deployed in the SDN directly without any licence or compatibility concern. Furthermore, data collection via SDN controllers is much easier than in the traditional network due to the centralised management. Sebbar *et al.* [66] deploy a SL model in the southbound interface (SBI) to detect MitM attack, it aims to disconnect a node if it is anomalous. The state of a network node, time to live and response time are the references to suspicious requests, the

TABLE 4. Anomaly detection in the cellular/wireless networks using ML.

Papers	ML Models	Anomaly	Methodology	Advantages	Disadvantages
Chen *et al.* [54]	SL: SVM, NB and DT	Malware	Capture mobile traffic to train a SL model for malware detection.	Perform well with highly imbalanced training data.	Performance is unstable when the imbalance ratio of dataset is under 1000.
Otoum *et al.* [55]	RL: Q-learning	Various attacks, such as DoS attacks	Use a RL model to analyse sensed data and detect malicious sensor behaviour.	High accuracy and detection rate.	The datasets used in the validation are too old.
Dromard *et al.* [56]	UL: density grid-based clustering	Various attacks, such as DDoS attacks	Split feature space of dataset, and run an UL model on each subspace to detect anomalies.	Detection delay is low, and high precision in the detection.	True negative and false negative rates are not mentioned.
Al *et al.* [57]	SL: KNN	Wide range of anomalies	Automatically annotate mobile traffic, and train a ML model from these labelled data to detect anomalies.	Save time for labelling data.	Recall and precision are not mentioned.
Dey *et al.* [58]	UL: K-Means	MitM	Use an UL model to cluster suspicious request based on inter-packet delay.	No rule update is required. The complexity of model is low, and can be customised.	As accuracy, recall and precision are not mentioned, the performance is unclear.
Hoang *et al.* [59]	UL: OCSVM, K-Means	Eavesdropping	Detect attacks using unsupervised clustering models with only one class of data tagged.	One-class labelled data provide reference to data clustering.	Accuracy is under 80% in some cases.
Qu *et al.* [60]	SL: SVM; UL: MSCA	Unknown attacks	Use an UL model to distinguish abnormal features, and a SL model to maximise the margin between the normal and abnormal patterns.	Misclassification is minimised via the combination of SL and UL models.	The dataset used in the validation is too old.
Vanerio *et al.* [61]	SL: Random Forest (RF), and *etc.*	Various attacks, such as DDoS attacks	Use ensemble method to obtain multiple predictions, and the final decision is made through these outputs.	Predictive performance is improved.	True positive rate is low in some cases.

data that collected via SBI are labelled normal or abnormal according to these conditions. Then labelled data are sent to the RF algorithm to train a classifier, which allows or drops new connection requests. Khamaiseh *et al.* [67] explore time-window of traffic in early detection using SL in SDN, as a small time-window means a short duration before SL making the decision, it could be a double-edged sword leading to an early detection or a worse accuracy, because the SL predictor may not have decent information to make the correct prediction. The centralised control of SDN allows the controller to periodically collect statistics from switches, so that the SL model can obtain up-to-date information to judge if a request is malicious. The predictor is trained offline with existing datasets, and then any new request is sent to the predictor for inspection. Based on the output, flow entries to forward or drop packets are inserted to the forwarding devices [68], [69].

Since SDN decouples the control and data plane from legacy networks, the link between these two planes are no longer sitting in the same hardware. Thus, attacks against control plane, data plane and the link between them must be considered separately. Santos *et al.* [70] compare the performance of SL models in DDoS attack detection in SDN, the target of DDoS attack includes three categories: controller, flow table and the bandwidth between the switch and controller. It is found from the test that the most important features for correct prediction are the IP source port, and the number of packets and bytes in flows.

Sometimes the outcome of ML model is not accurate enough, adding an extra step, such as entropy measurement, to double check the data could be an approach to improve the performance in anomaly detection. Dehkordi *et al.* [71] propose to collect statistics of network every period of time and calculate the entropy to see if it is normal, if the entropy value is under a predefined threshold, then a SL classifier is involved to determine whether it is an attack. Song *et al.* [72] introduce three subsystems running over the controller to predict threats in the SDN, these subsystems are used for data processing, classifier creation and decision making. As the training is based on past experience, data processing filters key information and discard irrelevant data so as to provide useful clues to the RF classifier. Based on the prediction from RF classifier, normal requests are processed and abnormal data are blocked. Furthermore, when the decision is ambiguous, the model will use the entropy to measure the ambiguity to make the final decision.

According to the user requirement, RL can be applied to trigger specific operations against intrusions. Now that the mitigation of DDoS attacks is to drop malicious traffic, Simpson *et al.* [73] introduce a probability value in the agent to instruct the switch to drop relevant packets. Two agent modes, instant and guarded, are proposed and validated in the single and multi-agent scenarios. In the instant mode, an agent directly chooses the probability of drop to partially discard current traffic flows, and it allows at least 10% flows to pass through. This mode has no interest in the future state. Then in the guarded mode, traffic could be completely blocked, and the future state may cause the update of state-action values. Through the evaluation, the instant agent mode performs well in the multi-agent scenario, in which several agents are working separately to protect the traffic towards

TABLE 5. Anomaly detection in the SDN using ML.

Papers	ML Models	Anomaly	Methodology	Advantages	Disadvantages
Sebbar *et al.* [66]	SL: RF	MitM	Use a SL model to classify connection request through SBI in SDN.	Detection of malicious node is accelerated, and time complexity is reduced.	The test is running in a controlled environment.
Khamaiseh *et al.* [67]	SL: SVM, NB, KNN	Saturation attacks	Train a SL model through the time-window of traffic to detect saturation attacks.	Unknown type of saturation attacks can be detected.	The detection performance is related to the test environment setup.
Alshamrani *et al.* [68], Akbacs *et al.* [69]	SL: SVM, NB, DT and KNN	Misbehaviour attacks, new flow attacks and DoS/DDoS attacks	Train a SL model offline, and collect traffic statistics via the controller for classification.	Easy to collect real time data and deploy ML models.	The datasets used in the validation are too old.
Santos *et al.* [70]	SL: SVM, DT, RF, Multiple Layer Perceptron (MLP)	DDoS attacks	Verify the performance of SL models in detecting DDoS attacks.	Find the most essential features in DDoS detection in SDN.	Recall and precision are not mentioned.
Dehkordi *et al.* [71]	SL: BayesNet, DT, Random Tree, Logistic Regression	DDoS attacks	Use entropy to find suspicious data, and train a SL model to further check.	Accuracy is improved by this two-step inspection.	The resource consumption of the tenfold classification method is not mentioned.
Song *et al.* [72]	SL: RF	DoS attacks,and *etc.*	Train a SL model offline to predict data, and calculate the entropy value to make final decisions if ambiguous.	Reduces uncertainty when make decisions with a small feature set.	The dataset used in the validation is too old.
Simpson *et al.* [73]	RL: Semi Gradient Sarsa	DDoS attacks	Train a RL model to drop packets through the analysis on the source-destination pair.	The action of RL model is taken per flow, and the mitigation method allows some legitimate traffic to pass through during attacks.	Detailed performance is not mentioned.
Sampaio *et al.* [74]	RL: Q-learning	Flooding attacks	Use a RL model to achieve load balancing on each link in the SDN.	Advanced route management without human intervention.	The amount of states will exponentially increase when more switches join the network.
Han *et al.* [75]	RL: Double Deep Q-networks and Asynchronous Advantage Actor-critic	Cyber attacks aiming at compromise network nodes.	Use a RL model to protect network nodes from being compromised via the node and link state. Employ adversarial training to protect the RL model.	Autonomous defence in SDN, and the attack against the RL model itself is mitigated.	It is based on the assumption that the attacker has compromised all the nodes in the transmission path.

the same server; while the guarded mode has a better output under single-agent scenario, in which an agent controls all the flows going to the server. Sampaio *et al.* [74] deploy RL in the SDN to achieve load balancing. They use the SDN controller to monitor the load on each link, any link with over 80% load is regarded as a high status and will trigger the agent in RL to modify the route. After the route update, a positive reward is given if there is no link with high status, otherwise the reroute action has a negative reward. This model can also be adopted to redirect malicious traffic. Han *et al.* [75] divide a network into nodes and links, and each node or link only has two states. A node is either normal or compromised, while a link is either turned on or off. The combination of node and link state reflects the current network state. Depends on the state, an action could be switch on/off nodes or links, or even doing nothing. Since the objective is to protect critical servers, the reward is characterised by the availability of server, the cost of mitigation and the number of reachable network resources. Assuming the attacker is aware of the RL agent and able to falsify the reward, the adversarial training is found capable of alleviating the impact during RL training. The ML applications in the SDN are summarised in Table 5.

D. IoT

Along with the large number of connected devices in IoT, the process of big data and unknown attacks becomes a main problem in the security domain. Since more portable devices join the network with very limited security features than ever before, hackers have more opportunities to commit flooding attacks, because the behaviour of a user is more arbitrary in the IoT context, and compromising an IoT device is much easier than hacking a firewall-equipped computer. ML, especially UL, shows its capability to detect unknown attacks with low computational resources. Both feature selection and dimensionality reduction aim to reduce the number of features, the difference is that original features are not changed in the feature selection. Nõmm and Bahşi [76] explore botnet attack detection with a small number of features using UL models. Rather than running a common model for all the IoT devices, a separate model which works for each device is

TABLE 6. Anomaly detection in the IoT network using ML.

Papers	ML Models	Anomaly	Methodology	Advantages	Disadvantages
Nõmm *et al.* [76]	UL: iForest and OCSVM	Botnet attacks	Use UL models with less than 10 selected features to detect attacks.	Reduced feature set consumes less resources during process, while the detection rate is still reasonable.	Detection rate is low in some cases.
Liu *et al.* [77]	UL: K-Means	Packet modification attacks	Use an UL model to cluster nodes into three groups, and only remove highly suspicious node.	Higher accuracy than clusters.	Accuracy drops when the number of hops increases.
Karimipure *et al.* [78]	UL: Dynamic Bayesian Networks	False data injection attacks	The interaction in the subsystems is changed by attacks, as well as the pattern in the UL model.	Efficient computation and can detect unobservable attacks.	Recall and precision are not mentioned.
Ahmed *et al.* [79]	UL: iForest and PCA	Covert data integrity assaults	Use an UL model to create multiple forests, and the measurement with shortest average path length is attack.	Computational complexity is low and detection time is short.	Accuracy is low in some cases.
Ali *et al.* [80]	UL: AE; SL: Linear SVM	DDoS attacks	Extract and merge key features in a weighted fashion, then use a SL model to detect DDoS attacks.	Prediction accuracy is improved after feature learning.	Accuracy is low in some cases.
Bhatia *et al.* [81]	UL: AE	DDoS attacks	Train an UL model with normal traffic, and use it to detect attacks.	More effective against new and unknown attacks comparing to some SL models.	Only TCP traffic is considered in the model.
Anthi *et al.* [82]	SL: DT	12 kinds of attacks, such as DoS and MitM	Three-tier inspection using SL models to classify the attacks.	Can predict the type of attack.	Accuracy drops when validated by the unseen dataset.
Rathore *et al.* [83]	SSL: ELM, semi-supervised fuzzy C-means	DoS attacks, and *etc.*	Pseudo label unknown data by checking it twice, and classify it when both outputs have a high confidence.	Better labelling approach, and fast real time detection.	Accuracy is lower than 90%.
Li *et al.* [84]	SSL: Tri-training	Flooding attacks, and *etc.*	Train three classifiers with labelled data, and classify unlabelled data by majority voting.	Higher detection rate and lower error rate comparing to some SL models.	Insider attack is not considered in the proposed collaborative intrusion detection system.
Ravi *et al.* [85]	SSL: NN and K-Means	Data deluge attacks	Use the shortest Euclidean distance from the unknown data to the known clusters to annotate unlabelled data.	Higher accuracy than some SL models.	The dataset used in the validation is too old.
Gu *et al.* [86]	RL: Q-learning	High-rate and low-rate IoT attacks, such as ARP spoofing	Use a RL model to adjust detection threshold so as to improve the performance.	Well adaption to IoT environment.	How the detection rate obtained from tests is not elaborated.

proposed. Several main features are first selected by feature selection methods, such as entropy and variance, then these data are sent to the UL models for clustering, and it obtains an acceptable accuracy. To achieve a higher detection accuracy, Liu *et al.* [77] propose to cluster data into three groups rather than two based on the suspicious level, and only the highly dubious group is malicious. A node collects periodic probe messages from a trusted source node via multiple paths, and the contribution of each node to a path is calculated. These contribution metrics are used as features in the clustering. While Karimipure *et al.* [78] use a similar concept to partition the smart grid network into sub-systems, in which data are processed in parallel. And the behaviour in each sub-system is learnt to be the reference for anomaly detection.

Since the UL is capable of clustering data into groups without any training, it is also applied with other algorithms to detect anomalies in the IoT network. Ahmed *et al.* [79] utilise iForest algorithm to determine if a measurement sample in the smart grid network is compromised. To categorise samples, principal component analysis (PCA) is invoked to transform the data size to a smaller dimension first. Then, the iForest sets up a binary search tree to isolate each sample. As the UL model splits all the samples to groups, the samples who are easy to isolate are more likely to be abnormal. Because the amount of compromised sample is usually small, and its feature is different from normal samples.

The hybrid of UL and SL could also improve the efficiency of detection. Ali *et al.* [80] train auto-encoders (AE) via UL to extract the features from the unlabelled dataset. Next, these features are merged according to their weights. Finally, the combined features are computed in a supervised manner to create a detection model. It is worth noting that the training of AE here is different from the training in the SL, because AE does not use labelled data, its objective is to minimise the reconstruction error. The reconstruction error is defined as the difference between the original data and the reconstructed data, and this error devises a threshold that is used to classify the data. Bhatia *et al.* [81] also demonstrate AE based classifier, which is trained by only normal traffic, to detect DDoS attacks.

Despite the fact that a predictor is able to distinguish attack data from normal data, as well as to classify the attack type through training, employing a SL model for a specific job is also applicable. Anthi *et al.* [82] propose a three layer ML model, layer 1 for profiling and learning the normal behaviour of each device, layer 2 for anomaly detection and layer 3 for attack classification. Each layer has a SL model to make predictions, which means a specific type of attack is identified after being inspected three times.

Due to the significance of annotating untagged data in the SSL, Rathore and Park [83] propose a two-tier verification approach to classify unlabelled data. They use extreme learning machine (ELM) algorithm to train the model with classified data, and send both labelled and unlabelled data to semi-supervised fuzzy C-means to filter high confident unlabelled data. After that these unclassified data with high confidence are examined again using the model trained by ELM, only those still have a high confidence will then be allowed to join the labelled data group. This process recurs till all the data are classified. In order to gain high confidence when tagging unlabelled data, vote is also a quite popular solution. Li *et al.* [84] adopt disagreement-based principle in the tri-training [87] method to classify unlabelled data. When two learners agree on the classifier of a sample, but the third learner disagrees, then the third learner is taught by the previous two learners on this sample. Ravi and Shalinie [85] split labelled dataset into normal and various attack classes, samples from each class are picked in a stochastic way. The Euclidean distance of unlabelled data against these samples are calculated to find the minimum value, which classifies the unknown sample. This classification repeats multiple times, and a sample is labelled only after more than half of the decisions pointing to the same cluster.

Apart from detect anomalies directly, RL can also be applied to improve the existing solutions. Gu *et al.* [86] involve RL to adjust attack detection threshold in an entropy-based framework, it successfully improves the detection rate and decreases the false alarm rate. IoT related anomaly detection methods using ML are summarised in Table 6.

E. CLOUD/FOG/EDGE NETWORK

Filters and rules are popular anomaly detection measures in the legacy network, however, they haven't shown decent results in security investigation in the cloud/fog/edge networks. ML or the combination of ML and rules has produced satisfactory results when deployed as IDS in the cloud [88]. Kim *et al.* [89] design a hybrid ML model in the cloud environment to detect and classify network threats. Key features are first selected via RF algorithm, then unlabelled data are clustered by these key features using UL models, and these clusters are unnamed so far. In order to know what attack a cluster represents, a threat label is added to each sample for naming clusters later. The threat label is defined by the value of some features in the labelled data, and the cluster name is given by the distribution of threat labels in each cluster. Thus, UL and SL models are employed for anomaly detection and classification, respectively. Aljamal *et al.* [90] and Baek *et al.* [91] also employ UL models for clustering and SL models for training and detection. The new clusters are labelled based on the assumption that normal data are in the large and dense clusters while anomalies belong to small or sparse clusters. Thus, a threshold function is defined to judge whether a cluster is small or large, as well as its density. After that data in the small and sparse clusters are labelled as abnormal, and other data are tagged as normal. SL models are then trained by these labelled data and employed to detect anomalies in the network.

Salman *et al.* [92] categorise attack types by a step-wise model, it is an improvement from the traditional single-type model. A single-type model is trained by a specific type of attack with normal traffic, so that network traffic has to go through all the single-type models for attack categorisation. While the step-wise model divide normal and anomalous traffic first, and then SL models recognise attack types using anomaly data only. The step-wise model puts several attack types in the same group, and once the group is determined, the specific attack type is further detected. Chkirbene *et al.* [93], [94] split a time period into several slots and use a SL model to predict the network state within each time slot. The most frequent decision within the time period is chose to be the prediction. Moreover, a weight value is involved in the prediction phase to improve the accuracy, it is the bias in the prediction of each category. The final decision is made from the number of predictions under the same category multiply by its weight value. In other words, a prediction that has a high weighted category is more likely to influence the result. Priyadarshini *et al.* [95] use SDN controllers in the fog network to block DDoS attacks before they enter the cloud network. A classifier model is trained in advance and running over the SDN controller to determine the state of network traffic, and flow entries are generated by the controller and inserted to the switches in the network to forward legitimate packets and drop malicious packets. Apart from traffic analysis, SL can also be deployed to classify cloud users via their credentials, and access control is applied to the cloud network according to the user [96].

To obtain the finest ML model with a specific training set whilst averting overfitting, more than one model can be trained and compared to find the best one [97]. Xu *et al.* [98] first randomly choose samples from labelled dataset to create several subsets, and use bagging method to train multiple models from these subsets. Then another dataset which consists of unreliable anomalous and unlabelled data is employed to train another model. Also, various sampling ratio is verified to get the optimal ratio through evaluation metrics. Finally, all the models are assessed through the entire dataset, including both labelled and unlabelled data. Finally, the one with lowest learning error or balanced accuracy is the best fitted model.

SSL is also applied in the cloud and fog network due to the lack of labelled data. Xu *et al.* [99] introduce fog enabled infrastructure and fog assisted AI engine to deploy SSL models in the fog network. The infrastructure creates

TABLE 7. Anomaly detection in the Cloud/Fog/Edge networks using ML.

Papers	ML Models	Anomaly	Methodology	Advantages	Disadvantages
Kim *et al.* [89]	SL: RF; UL: K-Means and DB-SCAN [101]	Various network attacks.	Feature selection with a SL model and use an UL model to cluster unlabelled data. Define a cluster by the distribution of threat labels in the cluster.	Only needs a small amount of labelled data.	Validated by datasets rather than implemented in the real network.
Aljamal *et al.* [90], Baek *et al.* [91]	SL: SVM, NB, RF and Adaboosting; UL: K-Means	Various attacks, such as DoS attacks	Use an UL model for clustering and define a threshold to name the clusters. Use a SL model for traffic classification.	Automatic data tagging by clustering.	Accuracy is under 90%.
Salman *et al.* [92]	SL: linear regression and RF	Various attacks, such as DoS attacks	Filter malicious traffic from normal traffic, and then classify attack types from malicious traffic.	Accuracy of attack classification is improved.	Several attacks, including DoS attacks, cannot be clearly categorised.
Chkirbene *et al.* [93], [94]	SL: DT	Various attacks, such as DoS attacks	Use a SL model to predict network state per time slot, and classify data based on the votes during multiple time slots.	Accuracy of attack classification is improved.	Several attacks, including DoS attacks, cannot be clearly categorised.
Priyadarshini *et al.* [95]	SL: KNN, SVM and NB	DDoS attacks	Use a SL model in the SDN controller to protect fog networks.	A feasible method when taking response time and resource utilisation of fog/cloud network into consideration.	Accuracy is under 90%.
Xu *et al.* [98]	SL: Bagged Tree	Various attacks, such as DoS attacks	Train multiple SL models and find the best performed model.	High accuracy in the detection.	The performance validated by UNSW-NB15 is not good enough.
Xu *et al.* [99]	SSL: OCSVM	Various attacks, such as DoS attacks	Divide original dataset to several subsets and use a SSL model to find the optimal sampling ratio of attack samples.	High accuracy in the detection.	Recall is under 90% for all the attacks.
Gao *et al.* [100]	SSL: NN and PCA	Various attacks, such as DoS attacks	Combine both SL and UL models, and use labelled data to correct the prediction of unlabelled data.	The ability to recognise new traffic pattern is enhanced, and the detection accuracy is increased.	Accuracy is under 85%.

multiple virtual machines, and each machine hosts a partitioned subset from the original dataset. These partitioned subsets are then uploaded to the AI engine to train detection models. SSL model is applied to the same subset to find the optimal learning model based on the accuracy of detection. Gao *et al.* [100] employ ensemble method to train a NN classifier with labelled data, and then this classifier predicts all the unlabelled data. The prediction is processed through fuzziness evaluation to extract valuable information, after that these pseudo-tagged data enter an ensemble system to double check the classification before being accepted as training set. The implementation of ML in the cloud/fog/edge network are summarised in Table 7.

F. EXPERIMENTAL DATASET

Apart from the aforementioned networks, some ideas have been proposed in general, not targeting a specific network type, and these methods have been evaluated through public datasets. Hosseini and Azizi [102] detect DDoS attacks by inspecting packets twice in the network. Classifiers are trained offline by existing datasets, in the meanwhile, essential features are extracted for the detection phase. To complete the transmission to the server, a packet is examined on both the client side and network side. On the client side, the state of packet is predicted by its essential features and a divergence test. As long as the packet is not considered as an attack, it is sent to the network proxy for further inspection, otherwise it is dropped. In the network proxy, an attack profile database contains all the known attack patterns, any packet matches a profile is discarded. Even if the attack is new to the database, it can still be detected via the trained classifiers, and its characteristic is recorded in the database for future detection. Gu *et al.* [103] propose a two-layer hierarchical ensemble model to detect anomalies. They first split the original dataset into heterogeneous training sets by fuzzy C-means clustering algorithm. Then several base classifiers are trained by these subsets. Their outputs are aggregated in a nonlinear manner, and are fed to an upper layer classifier to train a final model. The decision whether the traffic is an intrusion is made by the final model.

When labelled data is unavailable, AE can be used to capture the non-linear correlations in the data feature, and to find the latent representation which is insensitive to the variance of data to determine if anomaly happens. Nicolau *et al.* [104] introduce new regularisers to the AEs to push normal data to a small area whose centre is the origin. So that abnormal data are easy to be figured out, as they locate far away from the origin. Choi *et al.* [105] prepare three training sets which have different ratios of abnormal data to normal data, and use each of them to train four AE models. Each AE model produces key features of the training set, and these key features are employed to reconstruct the original dataset. If the

TABLE 8. Anomaly detection using ML with experimental datasets.

Papers	ML Models	Datasets	Anomaly	Methodology	Advantages	Disadvantages
Hosseini *et al.* [102]	SL: NB, KNN, DT, RF and MLP	NSL-KDD	DDoS attacks	Inspect packets twice to determine the state.	Can distinguish new attack types.	True negative and false negative rates are not mentioned.
Gu *et al.* [103]	SL: SVM	NSL-KDD, KDD 99 and Kyoto 2006+	Various attacks, such as DDoS attacks	A two-layer ensemble classifier for intrusion detection, outputs from base classifiers are combine non-linearly.	Robust performance in terms of accuracy and training speed.	The dataset used in the validation is too old.
Nicolau *et al.* [104]	UL: AE	CTU13, UNSW-NB15 and NSL-KDD	Various attacks, such as DoS attacks	Adjust AE models to push normal data close to the origin, and abnormal data far from the origin.	Work efficiently on high dimensional data scenario, and minimise the effect of AE model selection.	Recall is under 90% in some cases.
Choi *et al.* [105]	UL: AE	NSL-KDD	Various attacks, such as DoS attacks	Involve four AE models to reduce features and reconstruct the original dataset, and the model classifies anomalous data from the reconstruction error.	High accuracy in the detection.	The dataset used in the validation is too old.
Ashfaq *et al.* [106]	SSL: NN	NSL-KDD	Various attacks, such as DoS attacks	Use fuzziness value to group unlabelled data, and only add the data with a high or low fuzziness value to the new training set.	Classification accuracy is improved.	Accuracy is under 85%.
Idhammad *et al.* [107]	SSL: co-clustering and extra trees	NSL-KDD, UNB ISCX 12 and UNSW-NB15	DDoS attacks	Use UL models to preprocess and classify data, and use a SL model for further classification.	Accuracy and false positive rate in detection are satisfactory.	Accuracy is unstable when validated by UNSW-NB15, it varies from 66% to 100%.
Zavrak *et al.* [108]	SSL: AE and VAE	CICIDS2017	Various attacks, such as DDoS attacks	Train UL models with SSL approach using normal data only, and validate these models via both normal and abnormal data.	Detection rate is acceptable with UL models.	The performance is not stable under various types of attacks.
Al-Jarrah *et al.* [109]	SSL: K-Means	NSL-KDD and Kyoto 2006+	Various attacks, such as DDoS attacks	Create multiple classifiers from labelled data to classify unlabelled data.	High performance in the detection with a low percentage of labelled data.	Testing time is too long.

reconstruction error is less than the threshold, it is normal data, otherwise, it is abnormal.

When the number of labelled data is small, how to annotate unlabelled data from these known ones has a deep impact on the performance of SSL models. Ashfaq *et al.* [106] introduce fuzziness to categorise unlabelled data into three groups, which are high, mid and low, with a model trained by NN, this model is initially created from labelled data and it gives each unlabelled data a fuzziness value. The data with a high or low fuzziness value will join the existing labelled data group, and this new group is used to train an updated model to classify the test data. While the mid fuzziness data group are still ambiguous according to the classifier, so they will not be added to the labelled data to reduce the risk of misclassification. Moreover, in the process of unlabelled data, Idhammad *et al.* [107] run four algorithms to reduce irrelevance and noise in the normal data to increase the accuracy in the DDoS attack detection. Entropy of Flow Size Distribution (FSD) within a time window is calculated and compared with threshold, an abnormal entropy triggers the traffic data in that specific time window be divided into three groups by co-clustering algorithm. Based on the assumption that attack traffic becomes much more than normal traffic during DDoS attacks, the group with a lower information gain ratio has the normal traffic only, and the other two groups contain anomalous traffic. After these unsupervised process, the two data groups with malicious traffic are sent to extra-tree algorithm for SL steps. With labelled normal data only, abnormal states can still be realised through SSL. Zavrak and Iskefiyeli [108] propose an AE based model whose training uses only normal data, after the model is trained the validation dataset, which is comprised of half normal and half anomalous data, is sent to the model to create a threshold for anomaly detection. Since a test sample will be rebuilt in a trained AE, the anomaly threshold is defined by the difference between the reconstructed and the original input data. If the difference is larger than the threshold, the sample is labelled as abnormal, otherwise it is normal. Al-Jarrah *et al.* [109] randomly divide the whole dataset into multiple clusters first, so that a cluster may contain labelled data only, or unlabelled data only, or mixed. For the cluster which has untagged data only, it finds the nearest labelled data to form a new mixed cluster, which contains both labelled and unlabelled data. Tri-training is employed to process mixed clusters, it creates three classifiers from the original dataset, and tags unlabelled data as long as two or three classifiers agree on the labelling. While for fully labelled cluster, the proposed model builds binary classifier if the cluster contains multiple classes of data, otherwise label the cluster with one class data only. The ML applications validated via public datasets are summarised in Table 8.

VI. CONCLUSION

Machine learning is trying to prove itself in multiple fields, among which anomaly detection is a feasible application that attracts lots of attentions. No matter what is the network scenario, people still have numerous options from ML models. Hence, we present a comprehensive review on the ML in network anomaly detection. From SL to RL, each category processes data in a different style, which leads to a large gap in the outcome. Supervised, unsupervised or semi-supervised learning model is picked based on the dataset on hand, the proportion of labelled data is a key factor in selecting a model. By contrast, RL is a totally different style, it allows the model to try all the state-action pairs so as to identify the best solution. In addition to the model selection, data quality is the most vital part for anomaly detection, it directly links to the prediction performance. Most of the solutions are validated by public datasets or in the simulation, it will be better to verify these models in the real network. And the resource consumption, such as training time and CPU utilisation, of the model is rarely discussed, this shall also be considered and studied to reflect the efficiency. In the future, we would like to explore more for the application of deep learning techniques in the next generation network, such as SDN and IoT.

ACKNOWLEDGMENT

The authors would like to thank Prof. Christopher Leckie, a Professor with the Department of Computing and Information Systems, The University of Melbourne, for the inspiring discussions and his insightful comments to make this paper more comprehensible.

REFERENCES

[1] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, Jan. 2016.

[2] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 303–336, 1st Quart., 2013.

[3] Y. Harbi, Z. Aliouat, S. Harous, A. Bentaleb, and A. Refoufi, "A review of security in Internet of Things," *Wirel. Pers. Commun.*, vol. 108, no. 1, pp. 325–344, Sep. 2019.

[4] J. C. C. Chica, J. C. Imbachi, and J. F. B. Vega, "Security in SDN: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 159, Jun. 2020, Art. no. 102595.

[5] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov, and M. Ylianttila, "Security for 5G and beyond," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3682–3722, 4th Quart., 2019.

[6] A. Handa, A. Sharma, and S. K. Shukla, "Machine learning in cybersecurity: A review," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 9, no. 4, p. e1306, 2019.

[7] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Proc. Comput. Sci.*, vol. 171, pp. 1251–1260, Jan. 2020.

[8] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2015.

[9] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," 2017, *arXiv:1701.02145*.

[10] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," *Comput. Netw.*, vol. 151, pp. 147–157, Mar. 2019.

[11] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Comput. Secur.*, vol. 81, pp. 123–147, Mar. 2019.

[12] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): A survey," *J. Netw. Comput. Appl.*, vol. 161, Jul. 2020, Art. no. 102630.

[13] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1686–1721, 3rd Quart., 2020.

[14] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," *J. Netw. Comput. Appl.*, vol. 153, Mar. 2020, Art. no. 102526.

[15] A. B. Nassif, M. A. Talib, Q. Nasir, H. Albadani, and F. M. Dakalbab, "Machine learning for cloud security: A systematic review," *IEEE Access*, vol. 9, pp. 20717–20735, 2021.

[16] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *Proc. IEEE 31st Comput. Secur. Found. Symp. (CSF)*, Jul. 2018, pp. 268–282.

[17] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2014.

[18] J. A. Morente-Molinera, J. Mezei, C. Carlsson, and E. Herrera-Viedma, "Improving supervised learning classification methods using multigranular linguistic modeling and fuzzy entropy," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 5, pp. 1078–1089, Oct. 2016.

[19] J. S. Angarita-Zapata, A. D. Masegosa, and I. Triguero, "A taxonomy of traffic forecasting regression problems from a supervised learning perspective," *IEEE Access*, vol. 7, pp. 68185–68205, 2019.

[20] G. Huang, S. Song, J. N. D. Gupta, and C. Wu, "Semi-supervised and unsupervised extreme learning machines," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014.

[21] X. J. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1530, 2005.

[22] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–130, 2009.

[23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[24] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised classification algorithms in machine learning: A survey and review," in *Emerging Technology in Modelling and Graphics*. Singapore: Springer, 2020, pp. 99–111.

[25] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.

[26] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Machine Learning Proceedings 1994*. Amsterdam, The Netherlands: Elsevier, 1994, pp. 121–129.

[27] A. Kannan, G. Q. Maguire, Jr., A. Sharma, and P. Schoo, "Genetic algorithm based feature selection algorithm for effective intrusion detection in cloud networks," in *Proc. IEEE 12nd Int. Conf. Data Mining Workshops*, Dec. 2012, pp. 416–423.

[28] S. Ustebay, Z. Turgut, and M. A. Aydin, "Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier," in *Proc. Int. Congr. Big Data, Deep Learn. Fighting Cyber Terrorism (IBIGDELFT)*, Dec. 2018, pp. 71–76.

[29] N. Sánchez-Marono, A. Alonso-Betanzos, and M. Tombilla-Sanromán, "Filter methods for feature selection—A comparative study," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.* Berlin, Germany: Springer, 2007, pp. 178–187.

[30] D. Aksu, S. Üstebay, M. A. Aydin, and T. Atmaca, "Intrusion detection with comparative analysis of supervised learning techniques and Fisher score feature selection algorithm," in *Proc. Int. Symp. Comput. Inf. Sci.* Cham, Switzerland: Springer, 2018, pp. 141–149.

[31] S. Su, Y. Sun, X. Gao, J. Qiu, and Z. Tian, "A correlation-change based feature selection method for IoT equipment anomaly detection," *Appl. Sci.*, vol. 9, no. 3, p. 437, Jan. 2019.

[32] X.-Y. Liu, Y. Liang, S. Wang, Z.-Y. Yang, and H.-S. Ye, "A hybrid genetic algorithm with wrapper-embedded approaches for feature selection," *IEEE Access*, vol. 6, pp. 22863–22874, 2018.

[33] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 703–715, May 2019.

[34] A. Paudice, L. Muñoz-González, A. Gyorgy, and E. C. Lupu, "Detection of adversarial training examples in poisoning attacks through anomaly detection," 2018, *arXiv:1802.03041*.

[35] D. K. K. Reddy, H. Behera, G. S. Pratyusha, and R. Karri, "Ensemble bagging approach for iot sensor based anomaly detection," in *Intelligent Computing in Control and Communication*. Singapore: Springer, 2021, pp. 647–665.

[36] A. Shahraki, M. Abbasi, and Ø. Haugen, "Boosting algorithms for network intrusion detection: A comparative evaluation of real AdaBoost, gentle AdaBoost and modest AdaBoost," *Eng. Appl. Artif. Intell.*, vol. 94, Sep. 2020, Art. no. 103770.

[37] T. Zoppi, A. Ceccarelli, L. Salani, and A. Bondavalli, "On the educated selection of unsupervised algorithms via attacks and anomaly classes," *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102474.

[38] I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," in *Proc. Int. Conf. Netw. Digit. Technol.* Berlin, Germany: Springer, 2012, pp. 135–145.

[39] D. S. Terzi, R. Terzi, and S. Sagiroglu, "Big data analytics for network anomaly detection from netflow data," in *Proc. Int. Conf. Comput. Sci. Eng. (UBMK)*, Oct. 2017, pp. 592–597.

[40] J. Meira, R. Andrade, I. Praça, J. Carneiro, V. Bolón-Canedo, A. Alonso-Betanzos, and G. Marreiros, "Performance evaluation of unsupervised techniques in cyber-attack anomaly detection," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 11, pp. 4477–4489, Nov. 2020.

[41] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, Feb. 2020.

[42] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommun. Syst.*, vol. 70, no. 3, pp. 447–489, 2019.

[43] S. Mavoungou, G. Kaddoum, M. Taha, and G. Matar, "Survey on threats and attacks on mobile networks," *IEEE Access*, vol. 4, pp. 4543–4572, 2016.

[44] T. Jafarian, M. Masdari, A. Ghaffari, and K. Majidzadeh, "A survey and classification of the security anomaly detection mechanisms in software defined networks," *Cluster Comput.*, vol. 24, no. 2, pp. 1235–1253, Jun. 2021.

[45] B. B. Zarpelão, R. S Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.

[46] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *J. Netw. Comput. Appl.*, vol. 77, pp. 18–47, Jan. 2017.

[47] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença, Jr., "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Syst. Appl.*, vol. 92, pp. 390–402, Feb. 2018.

[48] Y. Gu, K. Li, Z. Guo, and Y. Wang, "Semi-supervised K-means DDoS detection method using hybrid feature selection algorithm," *IEEE Access*, vol. 7, pp. 64351–64365, 2019.

[49] M. Alauthman, N. Aslam, M. Al-kasassbeh, S. Khan, A. Al-Qerem, and K.-K. R. Choo, "An efficient reinforcement learning-based botnet detection approach," *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102479.

[50] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decis. Support Syst.*, vol. 107, pp. 88–102, Mar. 2018.

[51] Y. Xu, N. Chen, H. Zhang, and B. Liang, "Adaptive anomaly detection strategy based on reinforcement learning," in *Proc. Int. Conf. Pioneering Comput. Scientists, Eng. Educators*. Singapore: Springer, 2018, pp. 493–504.

[52] K. Sethi, E. S. Rupesh, R. Kumar, P. Bera, and Y. V. Madhav, "A context-aware robust intrusion detection system: A reinforcement learning-based approach," *Int. J. Inf. Secur.*, vol. 19, no. 6, pp. 657–678, Dec. 2020.

[53] Q. Jin and L. Wang, "Intranet user-level security traffic management with deep reinforcement learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[54] Z. Chen, Q. Yan, H. Han, S. Wang, L. Peng, L. Wang, and B. Yang, "Machine learning based mobile malware detection using highly imbalanced network traffic," *Inf. Sci.*, vols. 433–434, pp. 346–364, Apr. 2018.

[55] S. Otoum, B. Kantarci, and H. Mouftah, "Empowering reinforcement learning on big sensed data for intrusion detection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[56] J. Dromard, G. Roudière, and P. Owezarski, "Online and scalable unsupervised network anomaly detection method," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 1, pp. 34–47, Mar. 2016.

[57] S. M. A. Al Mamun and J. Valimaki, "Anomaly detection and classification in cellular networks using automatic labeling technique for applying supervised learning," *Proc. Comput. Sci.*, vol. 140, pp. 186–195, Jan. 2018.

[58] S. Dey, Q. Ye, and S. Sampalli, "A machine learning based intrusion detection scheme for data fusion in mobile clouds involving heterogeneous client networks," *Inf. Fusion*, vol. 49, pp. 205–215, Sep. 2019.

[59] T. M. Hoang, N. M. Nguyen, and T. Q. Duong, "Detection of eavesdropping attack in UAV-aided wireless systems: Unsupervised learning with one-class SVM and K-means clustering," *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 139–142, Feb. 2020.

[60] H. Qu, Z. Qiu, X. Tang, M. Xiang, and P. Wang, "Incorporating unsupervised learning into intrusion detection for wireless sensor networks with structural co-evolvability," *Appl. Soft Comput.*, vol. 71, pp. 939–951, Oct. 2018.

[61] J. Vanerio and P. Casas, "Ensemble-learning approaches for network security and anomaly detection," in *Proc. Workshop Big Data Anal. Mach. Learn. Data Commun. Netw.*, Aug. 2017, pp. 1–6.

[62] M. Usama, J. Qadir, A. Raza, H. Arif, K.-L.-A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha, "Unsupervised machine learning for networking: Techniques, applications and research challenges," *IEEE Access*, vol. 7, pp. 65579–65615, 2019.

[63] V. Timcenko and S. Gajin, "Ensemble classifiers for supervised anomaly based network intrusion detection," in *Proc. 13rd IEEE Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2017, pp. 13–19.

[64] B. S. Bhati, C. S. Rai, B. Balamurugan, and F. Al-Turjman, "An intrusion detection scheme based on the ensemble of discriminant classifiers," *Comput. Electr. Eng.*, vol. 86, Sep. 2020, Art. no. 106742.

[65] P. Casas, P. Fiadino, and A. D'Alconzo, "Machine-learning based approaches for anomaly detection and classification in cellular networks," in *Proc. TMA*, 2016, pp. 1–8.

[66] A. Sebbar, K. Zkik, Y. Baddi, M. Boulmalf, and M. D. E.-C. E. Kettani, "MitM detection and defense mechanism CBNA-RF based on machine learning for large-scale SDN context," *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 1–20, Dec. 2020.

[67] S. Khamaiseh, E. Serra, Z. Li, and D. Xu, "Detecting saturation attacks in SDN via machine learning," in *Proc. 4th Int. Conf. Comput., Commun. Secur. (ICCCS)*, Oct. 2019, pp. 1–8.

[68] A. Alshamrani, A. Chowdhary, S. Pisharody, D. Lu, and D. Huang, "A defense system for defeating DDoS attacks in SDN based networks," in *Proc. 15th ACM Int. Symp. Mobility Manage. Wireless Access*, Nov. 2017, pp. 83–92.

[69] M. F. Akbaş, C. Güngör, and E. Karaarslan, "Usage of machine learning algorithms for flow based anomaly detection system in software defined networks," in *Proc. Int. Conf. Intell. Fuzzy Syst.* Cham, Switzerland: Springer, 2020, pp. 1156–1163.

[70] R. Santos, D. Souza, W. Santo, A. Ribeiro, and E. Moreno, "Machine learning algorithms to detect DDoS attacks in SDN," *Concurrency Computation, Pract. Exper.*, vol. 32, no. 16, p. e5402, Aug. 2020.

[71] A. B. Dehkordi, M. Soltanaghaei, and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," *J. Supercomput.*, vol. 77, pp. 1–33, Mar. 2020.

[72] C. Song, Y. Park, K. Golani, Y. Kim, K. Bhatt, and K. Goswami, "Machine-learning based threat-aware system in software defined networks," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2017, pp. 1–9.

[73] K. A. Simpson, S. Rogers, and D. P. Pezaros, "Per-host DDoS mitigation by direct-control reinforcement learning," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 1, pp. 103–117, Mar. 2019.

[74] L. S. R. Sampaio, P. H. A. Faustini, A. S. Silva, L. Z. Granville, and A. Schaeffer-Filho, "Using NFV and reinforcement learning for anomalies detection and mitigation in SDN," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 432–437.

[75] Y. Han, B. I. Rubinstein, T. Abraham, T. Alpcan, O. De Vel, S. Erfani, D. Hubczenko, C. Leckie, and P. Montague, "Reinforcement learning for autonomous defence in software-defined networking," in *Proc. Int. Conf. Decis. Game Secur.* Cham, Switzerland: Springer, 2018, pp. 145–165.

[76] S. Nõmm and H. Bahşi, "Unsupervised anomaly based botnet detection in IoT networks," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 1048–1053.

[77] X. Liu, M. Abdelhakim, P. Krishnamurthy, and D. Tipper, "Identifying malicious nodes in multihop IoT networks using dual link technologies and unsupervised learning," *Open J. Internet Things*, vol. 4, no. 1, pp. 109–125, 2018.

[78] H. Karimipour, A. Dehghantanha, R. M. Parizi, K.-K. R. Choo, and H. Leung, "A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids," *IEEE Access*, vol. 7, pp. 80778–80788, 2019.

[79] S. Ahmed, Y. Lee, H. Seung-Ho, and I. Koo, "Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2765–2777, Mar. 2019.

[80] S. Ali and Y. Li, "Learning multilevel auto-encoders for DDoS attack detection in smart grid network," *IEEE Access*, vol. 7, pp. 108647–108659, 2019.

[81] R. Bhatia, S. Benno, J. Esteban, T. V. Lakshman, and J. Grogan, "Unsupervised machine learning for network-centric anomaly detection in IoT," in *Proc. 3rd ACM CoNEXT Workshop Big DAta, Mach. Learn. Artif. Intell. Data Commun. Netw.*, Dec. 2019, pp. 42–48.

[82] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9042–9053, Oct. 2019.

[83] S. Rathore and J. H. Park, "Semi-supervised learning based distributed attack detection framework for IoT," *Appl. Soft Comput.*, vol. 72, pp. 79–89, Nov. 2018.

[84] W. Li, W. Meng, and M. H. Au, "Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments," *J. Netw. Comput. Appl.*, vol. 161, Jul. 2020, Art. no. 102631.

[85] N. Ravi and S. M. Shalinie, "Semisupervised-learning-based security to detect and mitigate intrusions in IoT network," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 11041–11052, Nov. 2020.

[86] T. Gu, A. Abhishek, H. Fu, H. Zhang, D. Basu, and P. Mohapatra, "Towards learning-automation IoT attack detection through reinforcement learning," in *Proc. IEEE 21st Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Aug. 2020, pp. 88–97.

[87] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, Nov. 2005.

[88] R. S. S. Kumar, A. Wicker, and M. Swann, "Practical machine learning for cloud intrusion detection: Challenges and the way forward," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 81–90.

[89] H. Kim, J. Kim, Y. Kim, I. Kim, and K. J. Kim, "Design of network threat detection and classification based on machine learning on cloud computing," *Cluster Comput.*, vol. 22, no. 1, pp. 2341–2350, Jan. 2019.

[90] I. Aljamal, A. Tekeoğlu, K. Bekiroglu, and S. Sengupta, "Hybrid intrusion detection system using machine learning techniques in cloud computing environments," in *Proc. IEEE 17th Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, May 2019, pp. 84–89.

[91] S. Baek, D. Kwon, J. Kim, S. C. Suh, H. Kim, and I. Kim, "Unsupervised labeling for supervised anomaly detection in enterprise and cloud networks," in *Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Jun. 2017, pp. 205–210.

[92] T. Salman, D. Bhamare, A. Erbad, R. Jain, and M. Samaka, "Machine learning for anomaly detection and categorization in multi-cloud environments," in *Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, Jun. 2017, pp. 97–103.

[93] Z. Chkirbene, A. Erbad, and R. Hamila, "A combined decision for secure cloud computing based on machine learning and past information," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[94] Z. Chkirbene, A. Erbad, R. Hamila, A. Gouissem, A. Mohamed, and M. Hamdi, "Machine learning based cloud computing anomalies detection," *IEEE Netw.*, vol. 34, no. 6, pp. 178–183, Nov. 2020.

[95] R. Priyadarshini, R. K. Barik, and H. Dubey, "Fog-SDN: A light mitigation scheme for DDoS attack in fog computing framework," *Int. J. Commun. Syst.*, vol. 33, no. 9, p. e4389, Jun. 2020.

[96] D. Praveena and P. Rangarajan, "A machine learning application for reducing the security risks in hybrid cloud networks," *Multimedia Tools Appl.*, vol. 79, nos. 7–8, pp. 5161–5173, Feb. 2020.

[97] V. Morfino and S. Rampone, "Towards near-real-time intrusion detection for IoT devices using supervised learning and apache spark," *Electronics*, vol. 9, no. 3, p. 444, Mar. 2020.

[98] S. Xu, Y. Qian, and R. Q. Hu, "Data-driven edge intelligence for robust network anomaly detection," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1481–1492, Jul. 2019.

[99] S. Xu, Y. Qian, and R. Q. Hu, "A semi-supervised learning approach for network anomaly detection in fog computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[100] Y. Gao, Y. Liu, Y. Jin, J. Chen, and H. Wu, "A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system," *IEEE Access*, vol. 6, pp. 50927–50938, 2018.

[101] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, vol. 96, no. 34, pp. 226–231.

[102] S. Hosseini and M. Azizi, "The hybrid technique for DDoS detection with supervised learning algorithms," *Comput. Netw.*, vol. 158, pp. 35–45, Jul. 2019.

[103] J. Gu, L. Wang, H. Wang, and S. Wang, "A novel approach to intrusion detection using SVM ensemble with feature augmentation," *Comput. Secur.*, vol. 86, pp. 53–62, Sep. 2019.

[104] V. L. Cao, M. Nicolau, and J. McDermott, "Learning neural representations for network anomaly detection," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 3074–3087, Aug. 2018.

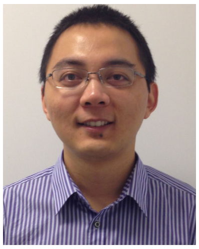
[105] H. Choi, M. Kim, G. Lee, and W. Kim, "Unsupervised learning approach for network intrusion detection system using autoencoders," *J. Supercomput.*, vol. 75, no. 9, pp. 5597–5621, Sep. 2019.

[106] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Inf. Sci.*, vol. 378, pp. 484–497, Feb. 2017.

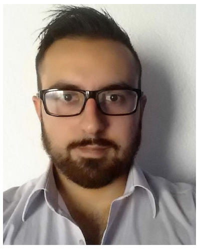
[107] M. Idhammad, K. Afdel, and M. Belouch, "Semi-supervised machine learning approach for DDoS detection," *Appl. Intell.*, vol. 48, no. 10, pp. 3193–3208, 2018.

[108] S. Zavrak and M. Iskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," *IEEE Access*, vol. 8, pp. 108346–108358, 2020.

[109] O. Y. Al-Jarrah, Y. Al-Hammdi, P. D. Yoo, S. Muhaidat, and M. Al-Qutayri, "Semi-supervised multi-layered clustering model for intrusion detection," *Digit. Commun. Netw.*, vol. 4, no. 4, pp. 277–286, Nov. 2018.



SONG WANG (Member, IEEE) received the B.E. degree in communication engineering from Shanghai University, China, in 2007, and the master's degree in telecommunication engineering from RMIT University, Australia, in 2016, where he is currently pursuing the Ph.D. degree in electrical and electronic engineering. His research interests include the security of software defined networks, machine learning, and the Internet of Things.



JUAN FERNANDO BALAREZO (Graduate Student Member, IEEE) received the Engineering degree in electronic and telecommunication engineering from the Army Polytechnic School, Ecuador, in 2013, and the master's degree in networks and security from Monash University, Australia, in 2016. He is currently pursuing the Ph.D. degree in electrical and electronic engineering. His research project is focused in security of software defined networks, denial of service attacks (DDoS), and attack modeling. He obtained the Postgraduate Dux Award from Monash University. In 2018, he obtained the Ethical Hacker Certification from the EC Council.

SITHAMPARANATHAN KANDEEPAN (Senior Member, IEEE) received the Ph.D. degree from the University of Technology Sydney. He has previously worked at the NICTA Research Laboratory (Canberra) and the CREATE-NET Research Center (Italy). He has authored over 140 peer-reviewed journals and conference papers, including a book on *Cognitive Radio Techniques* with Dr. A. Giorgetti. His current research interests include the Internet of Things (IoT), SDN, security, and wireless/mobile/satellite communications systems and networks. He was a recipient of the RMIT Research Excellence Award, in 2019, the IEEE Communications Society Certificate of Appreciation, in 2015, for ten years contribution to the field, and the IEEE Exemplary Reviewer Award, in 2011. He was the chair of several IEEE committees and workshops. He is an Editor for the Special Issue on Future Evolution of Public Safety Communications in the 5G Era in *ETT* journal (Wiley).

AKRAM AL-HOURANI (Senior Member, IEEE) received the B.Eng., M.B.A., and C.P.Eng. degrees and the Ph.D. degree from RMIT University, Melbourne, Australia, in 2016. He is currently a Senior Lecturer and the Program Manager of the Master of Engineering (telecommunication and networks), School of Engineering, RMIT University. He has published more than 55 journal articles and conference proceedings, including three book chapters. He has extensive industry/government engagement as a Chief Investigator in multiple research projects related to the Internet of Things (IoT), smart cities, and satellite/wireless communications. As a Lead Chief Investigator, he oversaw the design and deployment of the largest open IoT network in Australia in collaboration with five local governments "Northern Melbourne Smart Cities Network," this project has won the 2020 "IoT Awards," the official awards program of the IoT Alliance Australia. Prior his academic career, from 2006 to 2013, he had extensively worked in the ICT industry sector as an Research and Development Engineer, a Radio Network Planning Engineer, and an ICT Program Manager for several projects spanning over different technologies, including mobile networks deployment, satellite networks, and railway ICT systems. His current research interests include UAV communication systems, automotive and mmWave radars, energy efficiency in wireless networks, and the Internet of Things over satellite. In 2020, he has won the IEEE Sensors Council Paper Award for his contribution in hand-gesture recognition using neural networks. He is serving as an Associate Editor for *Frontiers in Space Technologies* and *Frontiers in Communications and Networks*. He is serving as a Guest Editor for the Special Issue "Satellite Communication" in *Remote Sensing* (MDPI).

KARINA GOMEZ CHAVEZ received the Engineering degree in electronic and telecommunication engineering from the National Polytechnic School, Ecuador, in 2006, the master's degree in wireless systems and related technologies from Turin Polytechnic, Italy, in 2006, and the Ph.D. degree in telecommunications from the University of Trento, Italy, in 2013. In 2007, she joined the Communication and Location Technologies Area, FIAT Research Centre. In 2008, she joined the Future Networks Area, Create-Net, working on several national, European, and industrial projects. In July 2015, she was a Lecturer at the School of Engineering, RMIT University, her role was to coordinate several networking courses and supervise several Ph.D. and master students. She is currently the Project Manager at Milano Teleport. She has several patents and has published her research in important journals and conferences. Her current research interests include energy efficiency networks, 4G/5G mobile networks architecture and network protocols, the Internet of Things (IoT) technologies, software defined networking (SDN), network functions virtualization (NFV), network security, multi-layer resources management, and orchestration and emergency communications.

BENJAMIN RUBINSTEIN (Member, IEEE) received the Ph.D. degree in computer science from UC Berkeley, in 2010. Prior to joining The University of Melbourne in 2013, he enjoyed four years in industry research labs, including Microsoft Research Silicon Valley and IBM Research Australia. He has been a part of teams that have analyzed privacy at the Australian Bureau of Statistics, a major bank, and transport for NSW; robustness of translation systems to data poisoning attacks with Facebook; helped identify and plug side-channel attacks against the Firefox browser; deanonymized Victorian Myki transport data and an unprecedented Australian Medicare data release; developed scalable Bayesian approaches to record linkage tested by U.S. Census; and shipped production systems for entity resolution in Bing and the Xbox360. Since joining Melbourne in 2013, he has been awarded $4.68m in competitive funding ($2.26m as lead). He also co-leads the CATCH Joint MURI-AUSMURI which has been funded over AUD $8m to convene a team of 16 experts across seven universities for fundamental discovery in robust human-AI teams for cybersecurity. His active research interests include machine learning, security and privacy, and databases, such as adversarial learning, differential privacy, and record linkage.