Cybersecurity

SURVEY Open Access

A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges

Ansam Khraisat* and Ammar Alazab*

Abstract

The Internet of Things (IoT) has been rapidly evolving towards making a greater impact on everyday life to large industrial systems. Unfortunately, this has attracted the attention of cybercriminals who made IoT a target of malicious activities, opening the door to a possible attack on the end nodes. To this end, Numerous IoT intrusion detection Systems (IDS) have been proposed in the literature to tackle attacks on the IoT ecosystem, which can be broadly classified based on detection technique, validation strategy, and deployment strategy. This survey paper presents a comprehensive review of contemporary IoT IDS and an overview of techniques, deployment Strategy, validation strategy and datasets that are commonly applied for building IDS. We also review how existing IoT IDS detect intrusive attacks and secure communications on the IoT. It also presents the classification of IoT attacks and discusses future research challenges to counter such IoT attacks to make IoT more secure. These purposes help IoT security researchers by uniting, contrasting, and compiling scattered research efforts. Consequently, we provide a unique IoT IDS taxonomy, which sheds light on IoT IDS techniques, their advantages and disadvantages, IoT attacks that exploit IoT communication systems, corresponding advanced IDS and detection capabilities to detect IoT attacks.

Keywords: Malware, Intrusion detection system, IoT, Anomaly detection, Machine learning, Deep learning, Internet of things, Attacks, IoT security

Introduction

Internet of Things (IoT) are interconnected systems of devices that facilitate seamless information exchange between physical devices. These devices could be medical and healthcare devices, driverless vehicles, industrial robots, smart TVs, wearables and smart city infrastructures; and they can be remotely monitored and regulated. IoT devices are expected to become more prevalent than mobile devices and will have access to the most sensitive information, such as personal information. This will result in increasing attack surface area and probabilities of attacks will increase. As security will be a vital supporting element of most IoT applications, IoT intrusion detection systems need also be developed to secure communications enabled by such IoT technologies (Granjal et al., 2015).

In the last few years, advancement in Artificial Intelligent (AI) such as machine learning and deep learning techniques has been used to improve IoT IDS (Intrusion Detection System). The current requirement is to do an up-to-date, thorough taxonomy and critical review of this recent work. Numerous related studies applied different machine learning and deep learning techniques through various datasets to validate the development of

* Correspondence: a.khraisat@federation.edu.au; aalazab@mit.edu.au
Federation University Australia, Federation University Australia, Ballarat, Australia

IoT IDS. But, it's still not clear that which dataset, machine learning or deep learning techniques are more effective for building an efficient IoT IDS. Secondly, the time consumed in building and testing IoT IDS is not considered in the evaluation of some IDSs techniques, despite being a critical factor for the effectiveness of 'online' IDSs (Khraisat et al., 2019a).

This paper provides an up to date taxonomy, together with a critical review of the significant research works on IoT IDSs up to the present time; and a classification of the proposed systems according to the taxonomy. It provides a structured and comprehensive overview of the existing IoT IDSs so that a researcher can become quickly familiar with the key aspects of IoT IDS. This paper also provides a critical review of machine learning and deep learning techniques applied to build IoT IDS. The detection techniques, validation strategies, deployment strategies are reviewed, along with several techniques used in each method. The complexity of different detection techniques, intrusion deployment strategy, and their evaluation techniques are discussed, followed by a set of suggestions identifying the best techniques, depending on the nature of the IoT IDS. Challenges for the current IoT IDSs are also discussed. Compared to previous survey publications (Khraisat et al., 2019a; Benkhelifa et al., 2018; Chaabouni et al., 2019; Zarpelao et al., 2017; Hindy et al., 2018) this paper presents a discussion on IoT techniques, IoT deployment strategy and IDS dataset problems which are of main concern to the research community in the area of IoT intrusion detection systems (IDS). Prior studies such as (Yang et al., 2017; Yar & Steinmetz, 2019) have not completely reviewed IoT IDSs in terms of the datasets, challenges and techniques. In this paper, we provide a structured and contemporary, wide-ranging study on IDS in terms of techniques, IoT attacks and datasets; and also highlight challenges of the IoT techniques and then make recommendations.

During the last few years, several surveys on IoT IDS have been published. Table 1 shows the IDS techniques and datasets covered by this survey and previous survey papers. The comparison that in this table discusses the contributions of each survey related to the develop intrusion detection system for IoT. The survey on intrusion detection systems and taxonomy by Axelsson (Axelsson, 2000) classified intrusion detection systems based on the detection methods. The highly cited survey by Debar et al. (Debar et al., 2000) surveyed detection methods based on the behaviour and knowledge profiles of the attacks. A taxonomy of IoT intrusion systems by Liao et al. (Liao et al., 2013a), has presented a classification of five subclasses with an in-depth perspective on their characteristics: Statistics-based, Pattern-based, Rule-based, State-based and Heuristic-based.

The highly cited survey by Alvarenga et al. (Zarpelao et al., 2017) reviews the IoT security issues in general. Attacks against IoT devices are not discussed in their studies, such as Denial of Service (DoS) Attack and attack on RPL (Routing Protocol for Low-Power and Lossy Networks). Critical Infrastructure such as power systems, transport, the internet, air traffic control, railways and power plants could all be disrupted by an IoT attacker. The authors reviewed intrusion detection in IoT, and they presented a great taxonomy to classify the IoT IDSs based on detection method, IDS placement strategy, and security threat and validation strategy. It was also indicated by Alvarenga et al. (Zarpelao et al., 2017) in 2017 that intrusion detection for IoT is still in an initial stage and that the existing IDSs do not enough for a wide variety of IoT attacks. This paper explored and discussed if the recent IoT IDSs are enough to deal with different IoT attacks.

Existing review articles (e.g., such as (Chaabouni et al., 2019; da Costa et al., 2019; Buczak & Guven, 2016; Lunt, 1988; Agrawal & Agrawal, 2015)) focus on intrusion detection techniques or dataset issue or type of computer attack and IDS evasion. No articles comprehensively reviewed IoT IDS, dataset problems, deployment strategies, IoT Intrusion techniques, and different kinds of attack altogether. In addition, the development of IoT IDS has been such that several different systems have been proposed in the meantime, and so there is a need for an up-to-date. The updated survey of the taxonomy of IoT IDS discipline is presented in this paper further enhances taxonomies given in (Khraisat et al., 2019a; Benkhelifa et al., 2018; Chaabouni et al., 2019; Liao et al., 2013a).

Given the discussion on prior surveys, this article focuses on the following:

- Classifying various kinds of IoT IDS based on intrusion techniques, deployment strategy, and validation strategy.
- Presenting a recent works effort to improve IoT security IDS.
- Taxonomy of IoT attacks.
- Discussion of available IDS datasets.
- The challenges of IoT IDS.

Intrusion detection in the internet of things

In this section, a review of the existing IDS research for IoT is presented. Each research was categorized by considering the following characteristics: IDS placement strategy, detection method, and validation strategy. Figure 1 shows the classification of IDS for IoT networks, while Table 1 provides some recent related research.

Table 1 Comparison of this survey and similar surveys: (✓: Topic is covered, ✖ the topic is not covered)

Survey	Intrusion Detection System Techniques							Attacks on IoT	Validation Strategy	Deployment Strategy	IoT Dataset
	SIDS	AIDS					Hybrid IDS				
		Supervised	Unsupervised	Semi-supervised	Ensemble methods	Deep Learning					
Lunt (Lunt, 1988)	✓	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖
Axelsson (Axelsson, 2000)	✓	✓	✖	✖	✖	✖	✖	✖	✖	✖	✖
Liao, et al. (Liao et al., 2013b)	✓	✓	✓	✖	✖	✖	✓	✖	✖	✖	✖
Agrawal and Agrawal (Agrawal & Agrawal, 2015)	✓	✓	✓	✓	✓	✖	✓	✖	✖	✖	✖
Buczak and Guven (Buczak & Guven, 2016)	✓	✓	✓	✖	✓	✖	✓	✓	✖	✖	✖
Zarpelao, et al. (Zarpelao et al., 2017)	✖	✓	✓	✖	✖	✖	✖	✓	✓	✓	✖
Khraisat, et al. (Khraisat et al., 2019a)	✓	✓	✓	✓	✓	✓	✖	✖	✖	✖	✓
This survey	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

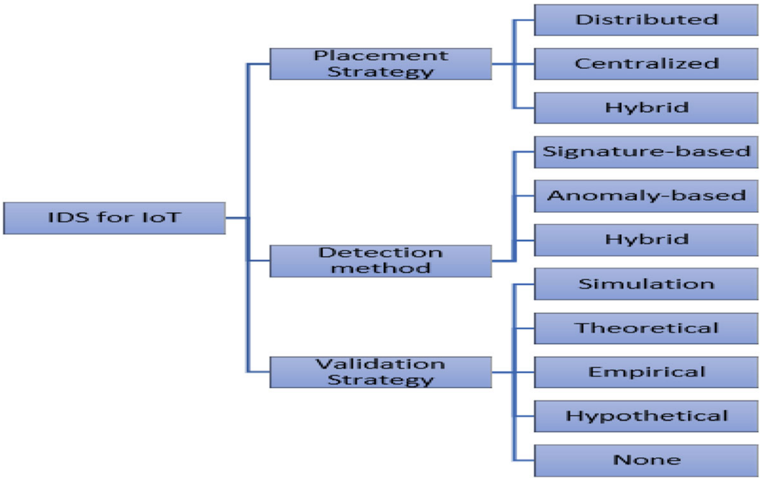

Fig. 1 Classification of IDSs for IoT

Figure 1 shows the IDS techniques, deployment strategy, validation strategy, attacks on IoT and datasets covered by this paper and previous research papers. The variety in the IoT IDS surveys indicates that a study of IDS for IoT must be reviewed. Specifically, none of these surveys cover all detection methods of IoT, which is considered crucial because of the heterogeneous nature of the IoT ecosystem. For that reason, this survey review IDS for IoT from a broad technological scale.

IoT intrusion detection systems methods

IoT Intrusion is defined as an unauthorised action or activity that harms the IoT ecosystem. In other words, an attack that results in any kind of damage to the confidentiality, integrity or availability of information is considered an intrusion. For example, an attack that will make the computer services unavailable to its legitimate users is considered an intrusion. An IDS is defined as a software or hardware system that maintains the security of the system by identifying malicious activities on the computer systems (Liao et al., 2013a). The main aim of IDS is to identify unauthorised computer usage and malicious network traffic which is not possible while using a traditional firewall. This results in making the computer systems highly protective against the malicious actions that compromise the availability, integrity, or confidentiality of computer systems. IDS system has two main sub-categories: Signature-based Intrusion Detection System (SIDS) and Anomaly-based Intrusion Detection System (AIDS).

Signature-based intrusion detection systems (SIDS)

Signature intrusion detection systems (SIDS) utilize pattern matching techniques to find a known attack; these are also known as Knowledge-based Detection or Misuse Detection (Khraisat et al., 2018). In SIDS, matching methods are used to find a previous intrusion. In other words, when an intrusion signature matches the signature of a previous intrusion that already exists in the signature database, an alarm signal is triggered. For SIDS, the host's logs are inspected to find sequences of commands or actions which have previously been identified as malware. SIDS has also been labelled in the literature as Knowledge-Based Detection or Misuse Detection (Modi et al., 2013).

Figure 2 demonstrates the conceptual working of SIDS approaches. The main idea is to build a database of intrusion signatures and to compare the current set of activities against the existing signatures and raise the alarm if a match is found. For example, a rule in the form of "if: antecedent -then: consequent" may lead to "if (source IP address=destination IP address) then label as an attack ".

SIDS usually gives an excellent detection accuracy for previously known intrusions (Kreibich & Crowcroft, 2004). However, SIDS is unable to detect zero-day attacks as the database does not contain a matching signature until the signature of the new attack is extracted and stored. SIDS is employed in a number of common tools, such as Snort (Roesch, 1999) and NetSTAT (Vigna & Kemmerer, 1999).

Traditional methods of SIDS have difficulty in identifying attacks that span multiple packets as they examine network packets and perform matching against a database of signatures. With the increased sophistication of

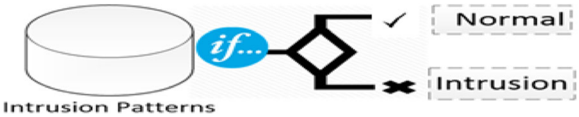

Fig. 2 Conceptual working of SIDS approaches

modern malware, extracting signature information from multiple packets may be required. With this, IDS needs to bring the contents of earlier packets as well. For creating a signature for SIDS, generally, there have been several methods where signatures are created as state machines (Meiners et al., 2010), formal language string patterns or semantic conditions (Lin et al., 2011).

With the increasing rate of zero-day attacks (Symantec, 2017), SIDS techniques have become progressively less effective because of the absence of signature for any such attacks. The other factors such as the polymorphic variants of the malware and the rising amount of targeted attacks also add up in compromising the adequacy of this traditional model. A potential solution to this problem would be to use AIDS techniques. AIDS works by differentiating between acceptable and unacceptable behaviour rather than profiling what is anomalous, as described in the next section, as described in the next section.

Anomaly-based intrusion detection system (AIDS)

AIDS has attracted a lot of scholars because of its feature to overcome the limitation of SIDS. In AIDS, a normal model of the behavior of a computer system is created using machine learning, statistical-based or knowledge-based methods. Any significant deviation between the observed behavior and the model is regarded as an anomaly, which can be interpreted as an intrusion. This kind of technique works on the fact that malicious behaviour is different from typical user behaviour. The behaviour of abnormal users that differentiates from the standard behaviour is defined as an intrusion. There are two phases in the development of AIDS: the training phase and the testing phase. In the training phase, the normal traffic profile is used to learn a model of normal behaviour. In the testing phase, a new data set is used to develop the system's capacity to generalise to previously unseen intrusions. AIDS can be sub-categorized based on the method used for training, for instance, statistical-based, knowledge-based and machine learning-based (Butun et al., 2014).

The main advantage of AIDS is the ability to identify zero-day attacks because recognizing the abnormal user activity does not rely on a signature database (Alazab et al., 2012). AIDS triggers a danger signal when the examined behavior deviates from normal behavior. Furthermore, AIDS has a number of benefits. First, they can discover internal malicious activities. If an intruder starts making transactions in a stolen account that are unidentified in the typical user activity, it creates an alarm. Second, it is challenging for a cybercriminal to recognize what is a normal user behavior without producing an alert as the system is constructed from customized profiles.

Table 2 presents the differences between signature-based detection and anomaly-based detection. The main difference between these two is that AIDS can discover zero-day attacks, whereas SIDS can only detect previously known intrusions. However, AIDS can result in a high false-positive rate because anomalies may just be new normal activities rather than genuine intrusions.

Since there is a lack of a taxonomy for anomaly-based intrusion detection systems, we have identified five subclasses based on their features: Statistics-based, Pattern-based, Rule-based, State-based and Heuristic-based as shown in Table 3.

Techniques for implementing AIDS

This section presents an overview of AIDS approaches proposed in recent years for improving detection accuracy and reducing false alarms.

AIDS methods can be categorized into four main groups: supervised learning (Chao et al., 2015), unsupervised learning (Elhag et al., 2015; Can & Sahingoz, 2015), reinforcement learning and deep learning (Buczak & Guven, 2016; Meshram & Haas, 2017). Supervised learning involves collecting and examining every input

Table 2 Comparisons of intrusion detection methodologies

		Advantages	Disadvantages
Detection methods	**SIDS**	• Very useful in identifying intrusions with minimum false alarms (FA). • Promptly identifies the intrusions. • Superior for detecting the known attacks. • Simple design	• It needs to be updated frequently with a new signature. • SIDS is designed to detect attacks for known signatures. When a previous intrusion has been altered slightly to a new variant, then the system would be unable to identify this new deviation of a similar attack. • Unable to detect the zero-day attack. • Not suitable for detecting multi-step attacks. • Little understanding of the insight of the attacks
	AIDS	• It could be used to detect new attacks. • Could be used to create intrusion signature	• AIDS cannot handle encrypted packets, so the attack can stay undetected and can present a threat. • High false positive alarms. • Hard to build a normal profile for a very dynamic computer system. • Unclassified alerts. • It needs initial training.

Table 3 Detection methodology characteristics for IoT IDS

Detection Methodology	Examples	Characteristics
Statistics based: analyzes the network traffic using complex statistical algorithms to process the information.	Bhuyan, et al. (Bhuyan et al., 2014)	• Needs a large amount of knowledge of statistics • Simple but less accurate • Real-time
Pattern-based: identifies the characters, forms, and patterns in the data.	Liao, et al. (Liao et al., 2013a) Riesen and Bunke (Riesen et al., 2008)	• Easy to implement • A hash function could be used for identification.
Rule-based: uses an attack "signature" to detect a potential attack on the suspicious network traffic.	Hall, et al. (Hall et al., 2009)	• The computational cost of rule-based systems could be very high because rules need pattern matching. • It is very hard to estimate what actions are going to occur and when • It requires a large number of rules for determining all possible attacks. • The low false-positive rate • High detection rate
State-based: examines a stream of events to identify any possible attack.	Kenkre, et al. (Kenkre et al., 2015)	• Probabilistic, self-training • Low false positive rate.
Heuristic-based: identifies any abnormal activity that is out of the ordinary activity.	Abbasi, et al. (Abbasi et al., 2014) Butun, et al. (Butun et al., 2014)	• It needs knowledge and experience • Experimental and evolutionary learning

variable and an output variable and you use an algorithm to learn the normal user behaviour from the input to the output. The objective is to approximate the mapping function so well that when a new input record is collected that predicts the output variables for that record. On the other hand, Unsupervised learning tries to identify the requested actions from existing system data such as protocol specifications and network traffic instances where you only have input data and no corresponding output variables, while reinforcement learning methods enable an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences. In reinforcement learning, the aim is to obtain an appropriate action model that would maximize the total cumulative reward of the agent. Deep learning models are based on artificial neural networks, specifically convolutional neural networks (CNN)s. These four classes along with examples of their subclasses are shown in Fig. 3.

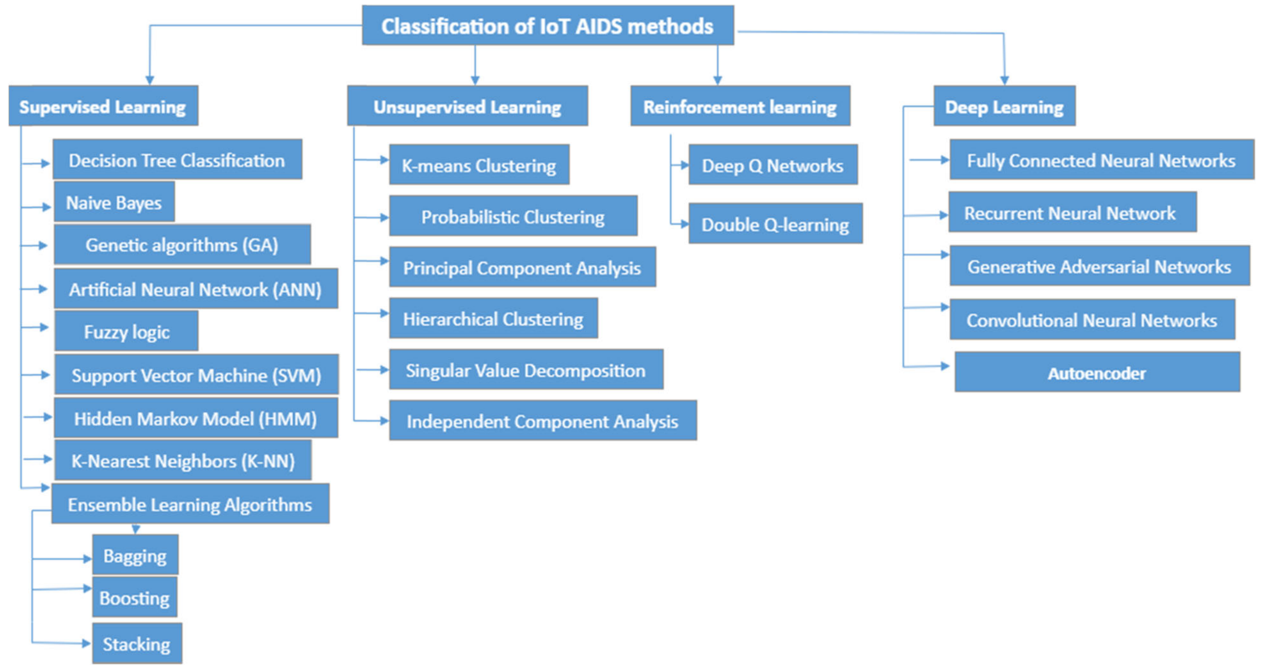
Classification of IoT AIDS methods
Supervised Learning
Decision Tree Classification
Naive Bayes
Genetic algorithms (GA)
Artificial Neural Network (ANN)
Fuzzy logic
Support Vector Machine (SVM)
Hidden Markov Model (HMM)
K-Nearest Neighbors (K-NN)
Ensemble Learning Algorithms
Bagging
Boosting
Stacking
Unsupervised Learning
K-means Clustering
Probabilistic Clustering
Principal Component Analysis
Hierarchical Clustering
Singular Value Decomposition
Independent Component Analysis
Reinforcement learning
Deep Q Networks
Double Q-learning
Deep Learning
Fully Connected Neural Networks
Recurrent Neural Network
Generative Adversarial Networks
Convolutional Neural Networks
Autoencoder

Fig. 3 Classification of AIDS methods

Machine learning is the process of extracting knowledge from large quantities of data. Machine learning models comprise of a set of rules, methods, or complex "transfer functions" that can be applied to find interesting data patterns or to recognise or predict behaviour (Dua & Du, 2016). Machine learning techniques have been applied extensively in the area of AIDS. To extract the knowledge from intrusion datasets, different algorithms and techniques such as clustering, neural networks, association rules, decision trees, genetic algorithms, and nearest neighbour methods are utilized.

Some prior research has examined the use of different techniques to build AIDSs. Chebrolu et al. examined the performance of two feature selection algorithms involving Bayesian networks (BN) and Classification Regression Trees (CRC) and combined these methods for higher accuracy (Chebrolu et al., 2005).

Bajaj et al. proposed a technique for feature selection using a combination of feature selection algorithms such as Information Gain (IG) and Correlation Attribute evaluation. They tested the performance of the selected features by applying different classification algorithms such as C4.5, naïve Bayes, NB-Tree and Multi-Layer Perceptron (Khraisat et al., 2018; Bajaj & Arora, 2013). A genetic-fuzzy rule mining method has been used to evaluate the importance of IDS features (Elhag et al., 2015). Thaseen et al. proposed NIDS by using the Random Tree model to improve accuracy and reduce the false alarm rate (Thaseen & Kumar, 2013). Subramanian et al. proposed classifying the NSL-KDD dataset using decision tree algorithms to construct a model for their metric data and studying the performance of decision tree algorithms (Subramanian et al., 2012).

Various AIDSs have been created based on machine learning techniques as shown in Fig. 4. The main aim of using machine learning methods is to create IDS that requires less human knowledge and improve accuracy. The quantity of AIDS which makes use of machine learning techniques has been increasing in the last few years. The main objective of IDS based on machine learning research is to detect patterns and build an intrusion detection system based on the dataset. Generally, there are two categories of machine learning methods, supervised and unsupervised.

Supervised learning in intrusion detection system

This subsection presents various supervised learning techniques for IDS. Each technique is presented in detail, and references to important research publications are presented.

Supervised learning-based IDS techniques detect intrusions by using labeled training data. A supervised learning approach usually consists of two stages, namely, training and testing. In the training stage, relevant features and classes are identified and then the algorithm learns from these data samples. In supervised learning IDS, each record is a pair, containing a network or host data source and an associated output value (i.e., label), namely intrusion or normal. Next, feature selection can be applied to eliminating unnecessary features. Using the training data for selected features, a supervised learning technique is then used to train a classifier to learn the inherent relationship that exists between the input data and the labelled output value. A wide variety of supervised learning techniques have been explored in the literature, each with its advantages and disadvantages. In the testing stage, the trained model is used to classify the unknown data into intrusion or normal class. The resultant classifier then becomes a model that, given a set of feature values, predicts the class to which the input data might belong. Figure 5 shows a general approach for applying classification techniques. The most existing IDSs proposed are trained in a supervised way. It implies that the cybersecurity professional need to label the network traffic and revise the model manually from time to time.

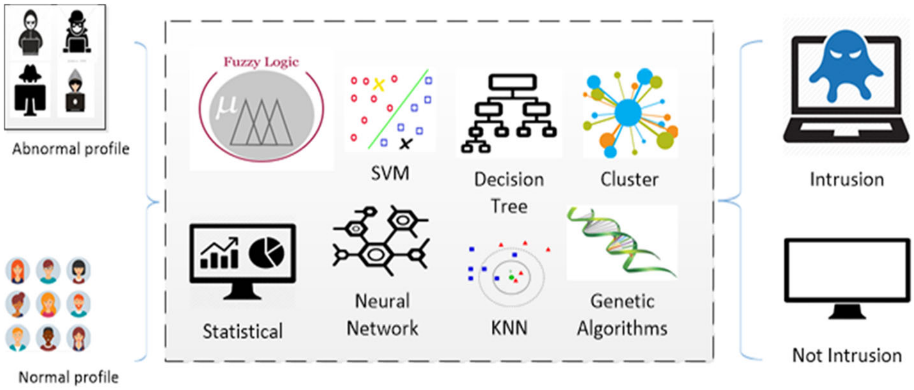

Fig. 4 Conceptual working of AIDS approaches based on machine learning

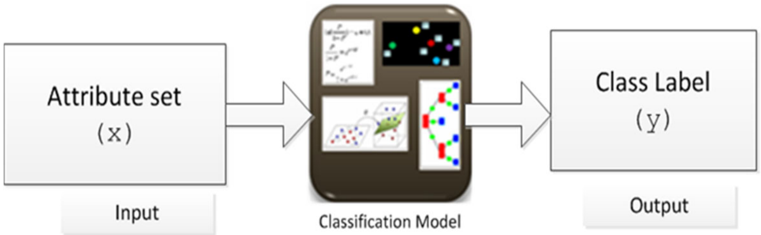

Fig. 5 Classification as the task

There are many classification methods such as decision trees, rule-based systems, neural networks, support vector machines, naïve Bayes and k-nearest neighbour. Each technique uses a learning method to build a classification model. However, a suitable classification approach should not only handle the training data, but it should also identify accurately the class of records it has not ever seen before. Creating classification models with reliable generalization ability is an important task of the learning algorithm.

Decision trees A decision tree has three basic components. The first component is a decision node, which is used to identify a test attribute. The second is a branch, where each branch represents a possible decision based on the value of the test attribute. The third is a leaf that comprises the class to which the instance belongs (Rutkowski et al., 2014). There are many different decision tree algorithms, including ID3 (Quinlan, 1986), C4.5 (Quinlan, 2014) and CART (Breiman, 1996).

Naïve Bayes This approach is based on applying Bayes' principle with robust independence assumptions among the attributes. Naïve Bayes answers questions such as "what is the probability that a particular kind of attack is occurring, given the observed system activities?" by applying conditional probability formulae. Naïve Bayes relies on the features that have different probabilities of occurring in attacks and normal behavior. The naïve Bayes classification model is one of the most prevalent models in IDS due to its ease of use and calculation efficiency, both of which are taken from its conditional independence assumption property (Yang & Tian, 2012). However, the system does not operate well if this independence assumption is not valid, as was demonstrated on the KDD'99 intrusion detection dataset, which has complex attribute dependencies (Koc et al., 2012). The results also reveal that the Naïve Bayes model has reduced accuracy for large datasets. A further study showed that the more sophisticated Hidden Naïve Bayes (HNB) model can be applied to IDS tasks that involve high dimensionality, extremely interrelated attributes and high-speed networks (Koc et al., 2012).

Genetic algorithms (GA) Genetic algorithms are a heuristic approach to optimization, based on the principles of evolution. Each possible solution is represented as a series of bits (genes) or chromosomes, and the quality of the solutions improves over time by the application of selection and reproduction operators, biased to favour fitter solutions. In applying a genetic algorithm to the intrusion classification problem, there are typically two types of chromosome encoding: one is according to clustering to generate binary chromosome coding method; another is specifying the cluster center (clustering prototype matrix) by an integer coding chromosome. Murray et al. have used GA to evolve simple rules for network traffic (Murray et al., 2014). Every rule is represented by a genome and the primary population of genomes is a number of random rules. Each genome is comprised of different genes that correspond to characteristics such as IP source, IP destination, port source, port destination and 1 protocol type (Hoque & Bikas, 2012).

Artificial neural network (ANN) ANN is one of the most broadly applied machine-learning methods and has been shown to be successful in detecting different malware. The most frequent learning technique employed for supervised learning is the backpropagation (BP) algorithm. The BP algorithm assesses the gradient of the network's error with respect to its modifiable weights. However, for ANN-based IDS, detection precision, particularly for less frequent attacks, and detection accuracy still need to be improved. The training dataset for less-frequent attacks is small compared to that of more-frequent attacks, and this makes it difficult for the ANN to learn the properties of these attacks correctly. As a result, detection accuracy is lower for less frequent attacks. In the information security area, huge damage can occur if low-frequency attacks are not detected. For instance, if the User to Root (U2R) attacks evade detection, a cybercriminal can gain the authorization privileges of the root user and thereby carry out malicious activities on the victim's computer systems. In addition, less common attacks are often outliers (Wang et al., 2010). ANNs often suffer from local minima and thus learning can become very time-consuming. The strength of ANN is that, with one or more hidden layers, it can produce highly nonlinear models that capture complex relationships between input attributes and classification labels. With the development of many variants such as recurrent and convolutional NNs, ANNs are powerful tools in many classification tasks including IDS.

Fuzzy logic This technique is based on the degrees of uncertainty rather than the typical true or false Boolean logic on which the contemporary PCs are created. Therefore, it presents a straightforward way of arriving at a conclusion based upon unclear, ambiguous, noisy, inaccurate or missing input data. With a fuzzy domain, fuzzy logic permits an instance to belong, possibly partially, to multiple classes at the same time. Therefore, fuzzy logic is a good classifier for IDS problems as the security itself includes vagueness, and the borderline between the normal and abnormal states is not well identified. In addition, the intrusion detection problem contains various numeric features in the collected data and several derived statistical metrics. Building IDSs based on numeric data with hard thresholds produces high false alarms. An activity that deviates only slightly from a model could not be recognized, or a minor change in normal activity could produce false alarms. With fuzzy logic, it is possible to model this minor abnormality to keep the false rates low. Elhag et al. showed that with fuzzy logic, the false alarm rate in determining intrusive actions could be decreased. They outlined a group of fuzzy rules to describe the normal and abnormal activities in a computer system, and a fuzzy inference engine to define intrusions (Elhag et al., 2015).

Support vector machines (SVM) SVM is a discriminative classifier defined by a splitting hyperplane. SVMs use a kernel function to map the training data into a higher-dimensioned space so that intrusion is linearly classified. SVMs are well known for their generalization capability and are mainly valuable when the number of attributes is large and the number of data points is small. Different types of separating hyperplanes can be achieved by applying a kernel, such as linear, polynomial, Gaussian Radial Basis Function (RBF), or hyperbolic tangent. In IDS datasets, many features are redundant or less influential in separating data points into correct classes. Therefore, feature selection should be considered during SVM training. SVM can also be used for classification into multiple classes. In the work by Li et al., an SVM classifier with an RBF kernel was applied to classify the KDD 1999 dataset into predefined classes (Li et al., 2012). From a total of 41 attributes, a subset of features was carefully chosen by using a feature selection method.

Hidden Markov model (HMM) HMM is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unseen data. Prior research has shown that HMM analysis can be applied to identify particular kinds of malware (Annachhatre et al., 2015). In this technique, a Hidden Markov Model is trained against known malware features (e.g., operation code sequence) and once the training stage is completed, the trained model is applied to score the incoming traffic. The score is then contrasted to a predefined threshold, and a score greater than the threshold indicates malware. Likewise, if the score is less than the threshold, the traffic is identified as normal.

K-Nearest Neighbors (KNN) classifier: The k-Nearest Neighbor (k-NN) technique is a typical non-parametric classifier applied in machine learning (Lin et al., 2015). The idea of these techniques is to name an unlabelled data sample to the class of its k nearest neighbors (where k is an integer defining the number of neighbors to be considered). Figure 6 illustrates a K-Nearest Neighbors classifier where k = 5. The point X represents an instance of unlabelled data that needs to be classified. Amongst the five nearest neighbors of X, there are three similar patterns from the class Intrusion and two from the class Normal. Taking a majority vote enables the assignment of X to the Intrusion class.

k-NN can be appropriately applied as a benchmark for all the other classifiers because it provides a good classification performance in most IDSs (Lin et al., 2015).

Ensemble methods Multiple machine learning algorithms can be used to obtain better predictive performance than any of the constituent learning algorithms alone (Vasan et al., 2020a). Training several classifiers at the same stage to detect different attacks, and then uniting their result to increase the detection rate. Typically, the ensemble's ability is better than a single classifier's, as it can enhance weak classifiers to produce better results than can a solitary classifier (Aburomman & Reaz, 2017). Several different ensemble methods have been proposed, such as Boosting, Bagging and Stacking.

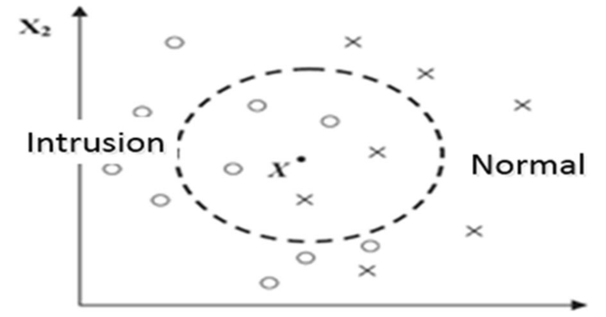

Fig. 6 An example of classification by k-Nearest Neighbour for k = 5

Boosting refers to a family of algorithms that can transform weak learners into strong learners. Bagging means training the same classifier on different subsets of the same dataset. Stacking combines various classification via a meta-classifier (Aburomman & Ibne Reaz, 2016). The base-level models are built based on a whole training set, and then the meta-model is trained on the outputs of the base level model as attributes.

Researchers have revealed that the combination of different classifier techniques is an effective way to resolve the shortcomings traditional IDSs have when they are applied for IoT. Jabbar et al. proposed an ensemble classifier that is built using Random Forest and also the Average One-Dependence Estimator (AODE which solves the attribute dependency problem in the Naïve Bayes classifier. Random Forest (RF) enhances precision and reduces false alarms (Jabbar et al., 2017). It is combining both approaches in ensemble results in improved accuracy over either technique applied independently.

More recently, Khraisat, et al. (Khraisat et al., 2019b) proposed a stacking ensemble method that combined the C5 decision tree classifier and one-class support vector machine. The reported classification accuracy of detection of malware is 94% on the IoT intrusion dataset C5 decision tree classifier, while it is 92.5% in stage two. They reported in the stacking ensemble, and the classification accuracy was 99.97%.

Unsupervised learning in intrusion detection system

Unsupervised learning is a kind of machine learning that makes use of input datasets without class labels to extract interesting information. The input data points are normally treated as a set of random variables. A joint density model is then created for the data set. In supervised learning, the output labels are given and used to train the machine to get the required results for an unseen data point. In contrast, in unsupervised learning, no labels are given, and instead, the data is grouped automatically into various classes through the learning process. In the context of developing an IDS, unsupervised learning means, use of a mechanism to identify intrusions by using unlabelled data to train the model. IoT network traffic is clustered into groups, based on the similarity of the traffic, without the need to predefine these groups.

As shown in Fig. 7, once records are clustered, all of the cases that appear in small clusters are labeled as an intrusion because the normal occurrences should produce sizable clusters compared to the anomalies. In addition, malicious intrusions and normal instances are dissimilar, thus they do not fall into an identical cluster.

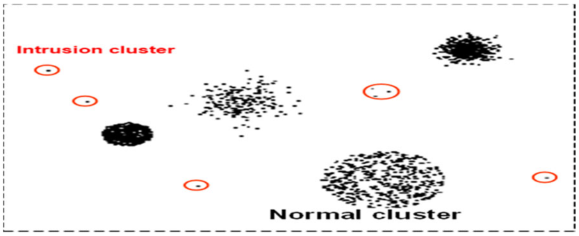

Fig. 7 Using Clustering for Intrusion Detection

K-means The K-means technique is one of the most prevalent techniques of clustering analysis that aims to separate 'n' data objects into 'k' clusters in which each data object is selected in the cluster with the nearest mean. K means it is an iterative clustering algorithm that aids to obtain the highest value for every iteration. It is a distance-based clustering technique and it does not need to compute the distances between all combinations of records. It applies a Euclidean metric as a similarity measure. The number of clusters is determined by the user in advance. Typically, several solutions will be tested before accepting the most appropriate one. Annachhatre et al. used the K-means clustering algorithm to identify different host behaviour profiles (Annachhatre et al., 2015). They have proposed new distance metrics that can be used in the k-means algorithm to relate the clusters closely. They have clustered data into several clusters and associated them with known behaviour for evaluation. Their outcomes have revealed that k-means clustering is a better approach to classify the data using unsupervised methods for intrusion detection when several kinds of datasets are available. Clustering could be used in IDS for reducing intrusion signatures, generate a high-quality signature or similar group intrusion.

Probabilistic clustering This technique uses probability distribution to create the clusters.

Principal component analysis is a common method for obtaining a set of low dimensional features from the largest set of features.

Hierarchical clustering This is a clustering technique that aims to create a hierarchy of clusters. Approaches for hierarchical clustering are normally classified into two categories:

(i) Agglomerative- bottom-up clustering techniques where clusters have sub-clusters, which in turn have sub-clusters and pairs of clusters are combined as one moves up the hierarchy.
(ii) Divisive - hierarchical clustering algorithms where iteratively the cluster with the largest diameter in

feature space is selected and separated into binary sub-clusters with a lower range.

Singular value decomposition It is a method of decomposing a matrix into other matrices as a series of linear approximations that expose the underlying meaning-structure of the matrix. The goal of Singular Value Decomposition is to uncover the optimal set of features that best predict the detection.

Independent component analysis It is used for showing hidden factors that underlie sets of random features.

A lot of work has been done in the area of the cyber-physical control system (CPCS) with attack detection and reactive attack mitigation by using unsupervised learning. For example, a redundancy-based resilience approach was proposed by Alcara (Alcaraz, 2018). He proposed a dedicated network sublayer that can handle the context by regularly collecting consensual information from the driver nodes controlled in the control network itself, and discriminating view differences through data mining techniques such as k-means and k-nearest neighbor. Chao Shen et al. proposed Hybrid-Augmented device fingerprinting for IDS in Industrial Control System Networks. They used different machine learning techniques to analyse network packets to filter anomaly traffic to detect intrusions in ICS networks (Shen et al., 2018).

Likewise, Khraisat, et al. (Khraisat et al., 2020) experimented with both single and ensemble classifiers composed of the decision tree, and SVM, for classification of the NLS KDD intrusion detection evaluation data set. They found that an ensemble of all three classifiers, based on majority voting, marginally out-performed all other classifiers.

Reinforcement learning

Deep Reinforcement learning utilizes deep learning and reinforcement learning principles for building IDS. Reinforcement learning involves an agent interacting with an environment. The agent is trying to achieve a goal of some kind within the environment. The purpose of the agent is to learn how to interact with its environment in such a way that allows it to achieve its goals.

Deep reinforcement learning is the application of reinforcement learning to train deep neural networks. It has an input layer, an output layer, and multiple hidden layers same as prior deep neural networks. However, our input is the state of the environment. For instance, a bus is trying to get its passengers to their destination. The inputs are the position, speed, and direction; our output is a series of possible actions like speed up, slow down, turn left, or turn right. In addition, we're feeding our rewards signal into the network so that we can learn to associate what actions produce positive results given a specific state of the environment.

Deep Q-network It is combined reinforcement learning and deep neural networks at scale. The algorithm was developed by enhancing a classic RL algorithm called Q-Learning with deep neural networks.

Double Q-learning It is an off-policy reinforcement learning algorithm that utilises double estimation to counteract overestimation problems with traditional Q-learning.

Deep learning

Deep learning is a form of machine learning where a computer uses a hierarchy of data based on experience and form multiple layers as an output. Deep learning can be supervised as well as unsupervised. In the case of supervised deep learning, data can be classified whereas in the case of unsupervised deep learning data patterns are analyzed. Deep learning is directly related to artificial intelligence where machines will acquire knowledge by learning with experience and will replace human intelligence. Deep learning works on the platform of artificial neural networks by studying massive amounts of data with the help of algorithms prepared by human intelligence. It is referred to as 'deep learning' as the artificial neural networks possess different deep layers that enables them to learn. Table 4 shows a Comparison of Machine learning and deep learning. Table 5 shows a summary of the deep learning model techniques.

Table 4 Comparison of Machine learning and deep learning

GENERAL	MACHINE LEARNING	Deep learning
Network Feature	Features extraction is required from the raw data to conduct a classification.	Features extraction is not necessary and the raw data could be used in a completely autonomous to build IDS.
Number of Contents	Only a part of available data is being utilized for building IDS. The data is scaled into a small vector of features, e.g. statistical correlations, it isinevitably throwing away most of the data	Processes all of the data, with a large number of features to detect the intrusions.
Correlations	Features selected by a human domain expert	Using raw data offers the capability to discover non-linear correlations between data that are too complex for a human expert.

Table 5 Summary of deep learning model techniques

Model	Learning Model	Input Data	Characteristics
FCNN	Supervised	Image, sound, etc.	- No special assumptions needed to be made about the input. -Requires a huge number of connections and network parameters.
RNN	Supervised	Serial, time-series	-Processes sequences of data through internal data. -Useful in IDS with time-dependent
GAN	Semi-supervised	various	-The GAN sets up a supervised learning problem to do unsupervised learning. -Less connection.
CNN	Supervised	Image, sound, etc.	-Need a large training dataset.
Autoencoder	Unsupervised	various	-It can be trained in an unsupervised manner. -It can be used for intrusion detection in the event of a poor reconstruction. - Generating new content - Filtering out noise

In neural networks, each neural node of every single hidden layer calculates the weighted values receiving from the previous layer and passes on the output values to the subsequent layer. The result value of the last layer can be considered as the final results achieved by the neural networks from the raw data.

Fully connected neural networks (FCNN) Fully Connected Feedforward Neural Networks are the standard network architecture applied in mainly basic neural network applications. Fully connected denotes that an individual neuron in the earlier layer is linked to every neuron in the subsequent layer. Feedforward indicates that neurons in any preceding layer are only ever connected to the neurons in a subsequent layer. Fully Connected Neural Networks can be used for feature extraction (Wang et al., 2020).

Recurrent neural network (RNN) The recurrent neural network can function efficiently on a series of data with variable input length. This means that RNNs use the information of its prior state as an input for their current prediction, and we can repeat this process for an arbitrary number of steps allowing the network to propagate information via its hidden state through time. This is essentially like giving a neural network a short-term memory. This feature makes RNNs very effective for working with sequences of data that occur over time. Yin, et al. (Yin et al., 2017) proposed a deep learning approach for intrusion detection using recurrent neural networks (RNN-IDS). their experimental results show that RNN-IDS is very appropriate for creating IDS with high accuracy and that its performance is superior to that of traditional machine learning classification methods in both binary and multiclass classification.

Generative adversarial networks (GAN) The Generative Adversarial Network is an integration of two deep learning neural networks: Generator Network, and a Discriminator Network. The Generator Network produces synthetic data, and the Discriminator Network tries to detect if the data that it's seeing is real or synthetic. These two networks are adversaries in the sense that they're both competing to beat one another.

Convolutional neural network (CNN) A Convolutional Neural Network is contained of one or more convolutional layers and then linked by one or more fully connected layers as in a standard multilayer neural network (Vasan et al., 2020b). A convolutional neural network contains an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically contain a sequence of convolutional layers that convolve with a multiplication. A CNN receives a 2-D input and abstracts high-level features via a sequence of hidden layers. CNN's, which better upon the architecture of the common neural networks, benefit from spatial features (Vasan et al., 2020c). Spatial features are usually applied types of traffic features in the area of IDS. When applying spatial features, network traffic is reformed into traffic images; it follows that the image classification technique is used to categorize the traffic images, which also ultimately achieves the objective of detecting the intrusion traffic. This technique is comparatively recent, but then numerous recent research results prove its great potential. For example, Vasan, et al. (Vasan et al., 2020b) adopted CNNS techniques and transformed the raw malware binary into both grayscale and colour images and apply the fine-tuned CNN architecture.

Autoencoder An autoencoder is trained to restructure its inputs. Autoencoders have been used for developing online IoT IDS (Mirsky et al., 2018). In general, an autoencoder trained on X gains the capability to restructure unobserved instances from the identical data distribution as X. If an instance does not be appropriate to the model learned from X, then it is expected the restructure to have a high error.

IoT IDS deployment strategies

IDS can also be classified based on the deployment used to detect IoT attacks. In IDS Deployment strategies, IDS can be classified as distributed, centralized or hybrid.

Distributed IDS

In distributed placement, the IoT devices could be responsible for checking other IoT devices. Distributed IDS be made up of several IDS over a big IoT ecosystem, all of which communicate with each other, or with a central server that assists advanced intrusion detection systems, packet analysis, and incident response.

Several IDS deploy distributed architectures. This includes a subset of the network checking the other nodes. Distributed IDS offers the incident analyst many advantages over centralized IDS. The main benefit is the capability to identify attack forms across a whole IoT ecosystem. This might increase prompt IoT attack prevention and detection. The additional supported benefit is to allow early detection of an IoT Botnet creating its way through corporate IoT devices. This data could then be used to detect and clean systems that have been infected by the IoT Botnet and stop further spread of the Botnet into the IoT ecosystem consequently take down any IoT devices damaged that would otherwise have occurred. Furthermore, the advantage of distributed IDS rather than centralized IDS computing resources also implies reduced control over those resources.

Centralized IDS

In the centralized IDS location, the IDS is placed in central devices, for instance, in the boundary switch or a nominated device. All the information that the IoT devices collect and then send to the network boundary switch passes through the boundary switch (Benkhelifa et al., 2018). Consequently, the IDS positioned in a boundary switch can check the packets switched between the IoT devices and the network. Despite this, checking the network packets that pass through the boundary switch is not adequate to identify anomalies that affect the IoT devices. The network traffic is monitored in centralized IDS. This traffic is extracted from the network through different network data sources such as packet capture, NetFlow, etc. The computers connected in a network can be monitored by Network-based IDS. Moreover, NIDS is also capable of monitoring the external malicious activities that could have been commenced from an external threat at an earlier stage, before these threats expand to other computer systems. However, NIDS comes with some limitations such as its restricted ability to inspect the whole data in a high bandwidth network because of the volume of data passing through modern high-speed communication networks (Bhuyan et al., 2014). NIDS deployed at several positions within a particular network topology, together with HIDS and firewalls, can provide a concrete, resilient, and multi-tier protection against both external and insider attacks. Table 6 shows a summary of comparisons between IDS deployment strategies.

Data source consists of system calls, application program interfaces, log files, data packets that are extracted from well-known attacks. These data sources can be useful to classify intrusion behaviors from abnormal actions.

Hierarchical IDS

In Hierarchical IDS, the network is separated into clusters. The sensor nodes that are adjacent to each other typically belong to the same cluster. Each cluster is assigned a leader, the so-called cluster head that screens the member nodes and plays a part in network-wide analyses.

IDS validation strategies

IDS Validation is the process for determining whether the IoT IDS model is an accurate enough representation of the system, for detecting IoT attacks. To validate the effectiveness of IDSs, researchers have used different techniques such as theoretical, empirical, and hypothetical strategies for validating their techniques.

There are many classification metrics for IDS, some of which are known by multiple names. Table 7 shows the confusion matrix for a two-class classifier which can be used for evaluating the performance of an IDS. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

IDS are typically evaluated based on the following standard performance measures:

- True Positive Rate (TPR): It is calculated as the ratio between the number of correctly predicted attacks and the total number of attacks. If all intrusions are detected then the TPR is 1 which is extremely rare for an IDS. TPR is also called a Detection Rate (DR) or the Sensitivity. The TPR can be expressed mathematically as

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (FPR): It is calculated as the ratio between the number of normal instances incorrectly classified as an attack and the total number of normal instances.

$$FPR = \frac{FP}{FP + TN}$$

- False Negative Rate (FNR): False negative means when a detector fails to identify an anomaly and

Table 6 Comparison of IDS deployment strategies based on their positioning

		Advantages	Disadvantages	Data source
IDS deployment strategies	Distributed IDS	• HIDS can check end-to-end encrypted communications behaviour. • No extra hardware is required. • Detects intrusions by checking the host file system, system calls or network events. • Every packet is reassembled • Looks at the entire item, not streams only	• Delays in reporting attacks • Consumes host resources • It needs to be installed on each host. • It can monitor attacks only on the machine where it is installed.	• Audits records, log files, Application Program Interface (API), rule patterns, system calls.
	Centralized IDS	• Do not impose an additional overhead on the sensor nodes. • Detects attacks by checking network packets. • Not required to install on each host. • Can check various hosts in the same period. • Capable of detecting the broadest ranges of network protocols	• IoT can be exposed if the centralized IDS is compromised. • Challenge is to identify attacks from encrypted traffic. • Dedicated hardware is required. • It supports only the identification of network attacks. • Difficult to analysis a high-speed network. • The most serious threat is the insider attack. • Not applicable For a large scale IoT ecosystem.	• Simple Network Management Protocol (SNMP) • Network packets (TCP/UDP/ICMP), • Management Information Base (MIB) • Router NetFlow records
	Hierarchical	• It uses NIDS, HIDS and wireless intrusion detection system (WIDS) presenting success in interoperability across heterogeneous Network types. • IDS is likely to be extremely deployable across big and heterogeneous IoT networks,	• the complexity of the IDS	Various

classifies it as normal. The FNR can be expressed mathematically as:

$$FNR = \frac{FN}{FN + TP}$$

- Classification rate (CR) or Accuracy: The CR measures how accurate the IDS is in detecting normal or anomalous traffic behavior. It is described as the percentage of all those correctly predicted instances to all instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Receiver Operating Characteristic (ROC) curve: ROC has FPR on the x-axis and TPR on the y-axis. In the ROC curve, the TPR is plotted as a function of the FPR for different cut-off points. Each point on the ROC curve represents an FPR and TPR pair corresponding to a certain decision threshold. As the threshold for classification is varied, a different point on the ROC is selected with different False Alarm Rate (FAR) and different TPR. A test with perfect discrimination (no overlap in the two distributions) has a ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity).

Table 7 Confusion matrix for IDS system

Actual Class	Predicted Class		
	Class	Normal	Attack
	Normal	True negative (TN)	False Positive (FP)
	Attack	False Negative (FN)	True positive (TP)

State-of-the-art intrusion detection in IoT

Table 8 shows a summary of the proposed research to IDSs for IoT.

Cho et al. proposed a methodology for checking packets that are passing through the border router for communication between physical and network devices. Their methodology is based on the botnet attacks by checking the packet length (Cho et al., 2009). However, no information is presented about the technique employed to create a normal behaviour profile. It is also not clear how the proposed IDS techniques would work on resource constraints nodes in the IoT.

Rathore et al. proposed semi-supervised Fuzzy learning-based distributed attack detection framework for IoT (Rathore & Park, 2018). The evaluation was done

Table 8 Summary of the proposed research to IDSs for IoT

Key References	Placement	Techniques	Security Threat	Validation Strategy
Cho, et al. (Cho et al., 2009)	Centralized	AIDS	Botnet	Simulation
Rathore and Park (Rathore & Park, 2018)	Distributed	AIDS	Network attack	Empirical (NSL-KDD Dataset)
Hodo, et al. (Hodo et al., 2016)	Centralized	AIDS	DoS attack	Simulation
Diro and Chilamkurti (Diro & Chilamkurti, 2018)	Distributed	AIDS	Network attack	Empirical (NSL-KDD Dataset)
Moustafa, et al. (Moustafa et al., 2019)	Distributed	Hybrid	The botnet, Man in the Middle	Empirical (UNSW-NB15)
Cervantes, et al. (Cervantes et al., 2015)	Distributed	Hybrid	Sinkhole attacks	Simulation
Khraisat, et al. (Khraisat et al., 2019b)	Distributed	Hybrid	IoT and network attacks	Empirical

on the NSL-KDD dataset and consequently suffered from the same limitations concerning the dataset as mentioned above.

Hodo et al. use an Artificial Neural Network (ANN) to detect DDoS and DoS attacks against legitimate IoT network traffic. The proposed ANN model was tested with the use of a simulated IoT network. Hoda et al. proposed a threat analysis of IoT using ANN to detect DDoS/DoS attacks. A multi-level perceptron, a type of supervised ANN, is trained using internet packet traces and then the model is assessed on its ability to thwart (DDoS/DoS) attacks (Hodo et al., 2016). Hoda et al. did not consider effectiveness after the deployment of the proposed IDS in the IoT ecosystem on low-capacity devices. According to their experimentation, the system achieved an accuracy of 99.4% for DDoS/DoS. However, no details of the dataset are provided.

Diro et al. developed an IoT network attack detection system based on distributed deep learning. Their work showed that distributed attack detection could identify IoT attacks better than a centralized strategy with a 96% detection rate. Their approach was evaluated using the NLS-KDD dataset. Even though this dataset is another version of the KDD data set, it still suffers from various issues reviewed by McHugh (McHugh, 2000). We believe this dataset should not be used as a practical benchmark dataset in the IoT as this data was collected from the traditional network (Diro & Chilamkurti, 2018). This leads us to develop IDSs that take into consideration the specific requirement of IoT protocol such as (Low-power Wireless Personal Area Networks) 6LowPAN. Hence, the Intrusion detection system that is created for the IoT ecosystem should operate under rigorous settings of low processing ability, high-speed connection, and big capacity data processing.

Moustafa et al. proposed an ensemble of IDSs to detect abnormal activities, in specific botnet attacks against Domain Name System (DNS), Hypertext Transfer Protocol (HTTP) and Message Queue Telemetry Transport (MQTT) (Moustafa et al., 2019). Their ensemble methods are based on the AdaBoost learning method and they used three machine learning techniques: Artificial Neural Networks (ANN), Decision Tree (DT) and Naive Bayes (NB) to evaluate their methodology (Moustafa et al., 2019). The proposed IDS result in significant overhead which degrades its performance.

Cervantes, et al. proposed IDS for detecting sinkhole attacks on 6LoWPAN for the IoT. Their IDS approach applies a combination of anomaly detection and support vector machine (SVM). IDS during the training process, each IDS agent trains the SVM and executes a majority voting decision to mark the infected nodes (Cervantes et al., 2015). Their simulation results show that their IDS achieve a sinkhole detection rate of up to 92% on the fixed scenario and 75% in a mobile scenario. However, their approach has not been evaluated for other types of attacks in the IoT.

Khraisat, et al. (Khraisat et al., 2019b) proposed an ensemble Hybrid Intrusion Detection System (HIDS) by combining a C5 classifier and a One-Class Support Vector Machine classifier. C5 classifier is used to detect well know intrusion. One-Class Support Vector Machine classifier is used to detect a new attack.

Attacks on IoT ecosystem

As IoT technology involves many devices like sensors, processors and many other technologies, the purpose of sharing the data and connecting to other networks has been served successfully. As it involves many devices connected, the data shared may not be secure and the security concern raises. IoT Security refers to protect the information shared among different networks through IoT devices using IoT technology. These devices are connected to others using the internet which allows vulnerabilities to take place by allowing the hacker to hack the data. Data without the security will lead to many concerns and brings huge loss for many industries and even to the individuals ending with the loss of the data from their systems (Khraisat et al., 2019b).

IoT grabbed the attention of the people and the organizations from many sectors onto it, by providing extreme benefits to them. Along with its tremendous growth, some security issues have risen by which IoT attacks have taken place by preventing people to use many of its upcoming applications. Hence, this section report discusses the concept of IoT security, the Challenge of IoT security, the impacts of them followed by the IoT attack and its types. IoT devices can be accessed from any place within a trusted network. So, there are chances of lots of malicious attacks in the IoT network. Hence, security, privacy, and confidentiality issues must be appropriately addressed in the IoT to protect it from malicious attacks. For example, the attacking of traffic lights and driverless vehicles not only reasons chaos and rises contamination, but also can initiate harm and severe collisions leading to wounded.

Different devices and equipment of home and office can be virtually connected with the help of the internet to left them they can perform their activities by monitoring the device's remote.

Figure 8 shows the IoT system architecture with layers where attacks can occur. An IoT system can comprise three fundamental layers which are the perception layer, network layer, and application layer (Liao et al., 2013a). The perception layer is the lowest layer of the conventional architecture of IoT. This layer consists of devices, sensors, and controllers. This layer's fundamental task is to gather valuable information from IoT sensors systems.

In the network layer, IoT involves a variety of diverse networks such as WSNs, wireless mesh networks, WLAN, etc. These networks help sensors in IoT exchange information. A gateway can simplify the communication of several sensors over the network. Thus, a gateway could be beneficial to handle many complex aspects involved in communication on the network. The network layer ensures the successful transmission of data while the application layer is the highest layer that processes the data for visualization.

In the application layer, the data source can be obtained from Internet Service Provider (ISP) and mobile network providers' web-based services, virtual online identities, edge network, devices logs, Radio-Frequency Identification (RFID) tags, and readers, etc.

Most of the attackers' target IoT devices and equipment rather than a single PC. IoT has an interconnection of

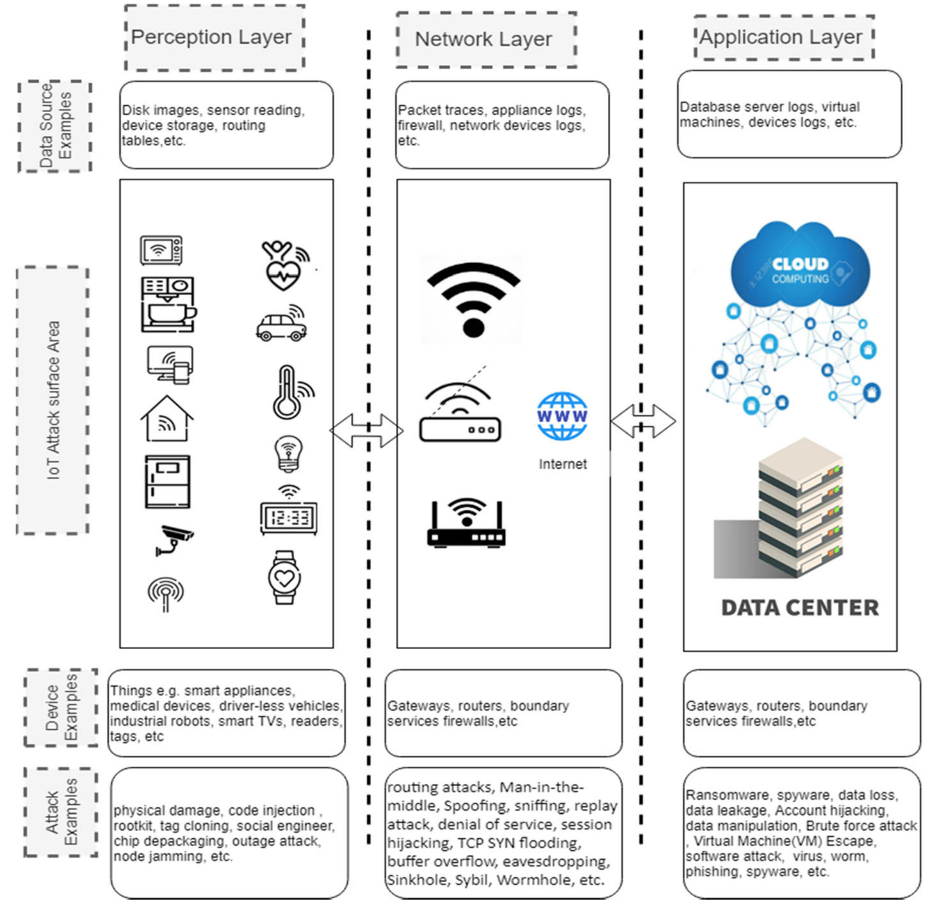
Perception Layer
Network Layer
Application Layer
Data Source Examples
Disk images, sensor reading, device storage, routing tables,etc.
Packet traces, appliance logs, firewall, network devices logs, etc.
Database server logs, virtual machines, devices logs, etc.
IoT Attack surface Area
12:33
Internet
CLOUD COMPUTING
DATA CENTER
Device Examples
Things e.g. smart appliances, medical devices, driver-less vehicles, industrial robots, smart TVs, readers, tags, etc
Gateways, routers, boundary services firewalls,etc
Gateways, routers, boundary services firewalls,etc
Attack Examples
physical damage, code injection , rootkit, tag cloning, social engineer, chip depackaging, outage attack, node jamming, etc.
routing attacks, Man-in-the-middle, Spoofing, sniffing, replay attack, denial of service, session hijacking, TCP SYN flooding, buffer overflow, eavesdropping, Sinkhole, Sybil, Wormhole, etc.
Ransomware, spyware, data loss, data leakage, Account hijacking, data manipulation, Brute force attack , Virtual Machine(VM) Escape, software attack, virus, worm, phishing, spyware, etc.

Fig. 8 IoT architecture and layer attacks

various devices and equipment along with some embedded devices as well. The major causes of IoT as a malware target can be summarized below:

- All the devices and equipment in an IoT need to be always on and it is easy for attackers to assess that equipment where the power mode is on at any point in time.
- Devices and equipment interconnected in an IoT are always connected and the attackers may access the interconnected devices from a single device.
- In most cases, proper security measures and knowledge to defend and tackle attack in a whole set of interconnected devices is difficult than in a single PC.
- Lack of proper encryption features in the interconnecting devices and weak passwords is another cause of malware target in IoT.
- The level of sophistication for the exploitation of the IoT is much lower and easy as compared to a single device.
- Twenty-four hours of internet exposure of the IoT devices and equipment is another cause of IoT as a malware target. Due to the unlimited internet connection, the devices will accept the incoming traffic signals and become vulnerable to attacks.

The attributes and features of malware differ in a single device and a set of interconnected devices and equipment.

Table 9 shows the different security attributes of a single device that is PC and the set of devices that is IoT about malware. Cyber-attacks on IoT applications can be both internal and external attacks. The attacker is a compromised node of the network in an inside attack whereas the attacker is not a part of the network in an outside attack. Figure 9 shows the significant types of cyber-attack that target IoT applications. The types of attacks as well as how the attack will impact the IoT network and their implications are described.

Table 9 Difference the security attributes between computer and IoT

Attributes	PC	IoT
Execution Platform Heterogeneity	Low	High
Malware family Variety	High	Low
Intrusion Detection Technique	Easy	Difficult
Internal Analysis	Easy	Very Difficult
Sandbox Execution	Easy	Difficult
Removing Malware	Medium	Very Difficult
Susceptibility Testing	Medium	Very Difficult

Physical/perception layer

Attacks are based on hidden aspects of devices and equipment. These attacks can take control of the device by tampering with hardware. IoT physical attacks are launched when an attack is close to the network or IoT device. Some of the significant threats at the physical/perception layer include:

Node tampering

Node Tampering refers to hacking the system to find the secret keys to decrypt the encrypted data.

Radio frequency (RF) Interface

Radio Frequency (RF) which is used for wireless communications among the IoT. This wireless technology for transmitting data between devices is vulnerable to several attacks that can easily damage the IoT devices.

Node jamming

Jamming attacks are a type of DoS attack where an adversary transmits a high-range signal to mess up the transmission. In Jamming Attacks, a malicious node in the sensor network broadcast a jamming signal which has a similar set of frequencies with the sensor nodes. This jamming attack stops the sensor nodes to transmit or accept data by creating a noise in the IoT network and making the services unavailable.

Node attack

The cybercriminal could get full control of sensor nodes. Because of the placement of IoT devices in different locations, tags are susceptible to physical attacks. A cybercriminal could simply take these tags and make a copy of them, which consider genuine tags to exploit an RFID system.

Physical damage

The attacker physically participates in the attack to modify the data or to steal confidential information.

Social engineering attacks

The attacker uses social engineering techniques to illegally access a system for installing malicious software secretly. IoT devices, particularly wearables, gather huge sizes of personally identifiable information (PII) to create a personalized experience for their customers. Such IoT gadgets also utilize the personal information of customers to bring user-friendly facilities, for example, ordering products online with voice control. However, PII can be attacked by cybercriminals to gain illegal access to sensitive information such as user passwords, purchase history, and personal information.

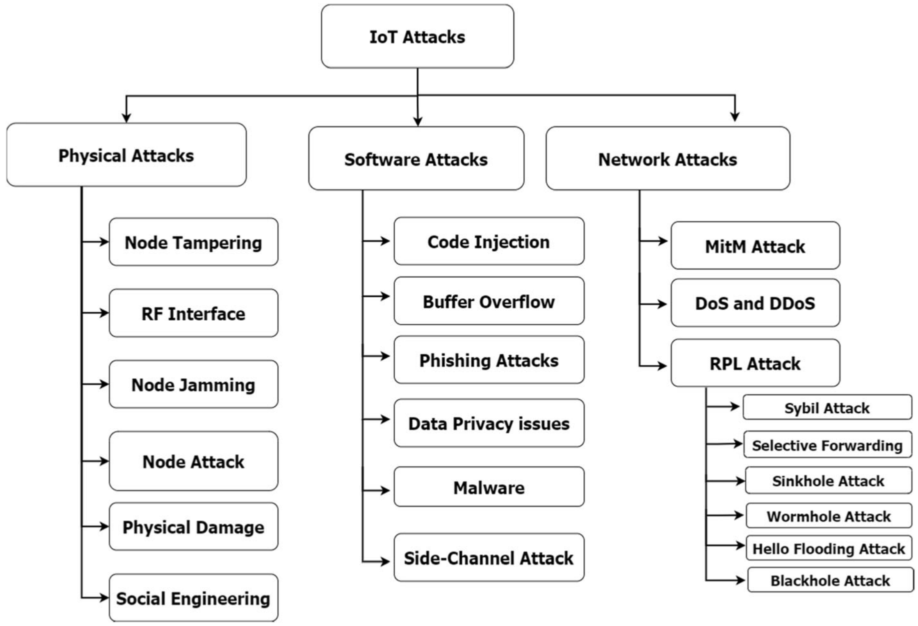

Fig. 9 Taxonomy of Security attacks within IoT

Software/ application layer

In IoT technology, the applications are developed using API's and these applications are web applications that cannot run without installing software. Software attacks are performed by using the software applications using phishing attacks, trojans, ransomware, worm, virus or by any other malicious content which may include spyware and adware.

Code injection

Inject code into a vulnerable sensor node and change the course of execution. For example, inaudible attack, DolphinAttack, inject inaudible voice commands at voice controllable systems by exploiting the ultrasound channel (Zhang et al., 2017). Another example, voice squatting attack in which the attacker manipulates the VPA service by injecting malicious code during the user's conversation with the service to steal her personal information.

Buffer overflow

A buffer overflow occurs when data are written to a sensor node buffer also corrupts data values in memory addresses adjacent to the destination buffer due to inadequate boundaries validating.

Data privacy issue

The Attackers may place RFID tags on many household items. Tracking IoT gadget using RFID tags could be used to threaten the privacy of users by tracking their activities and create a user profile.

Malware

Malware is any malicious software intended to produce harm or damage to IoT architecture. Broad diversity of malware forms exists, including viruses, worms, Trojan horses, ransomware, spyware, and adware.

Phishing attack

The attacker uses an IoT edge node as a trap. The goal is to collect information like passwords, usernames, etc.

Side-Channel attack

A side-channel attack breaks cryptography by using information disclosed by cryptography.

Network layer

Transmission of data takes place at the network layer where the security issues occur and may lead to the attacks taking place. These attacks may be Eavesdropping, man-in-the-middle attacks, DoS attacks, storage attacks, exploit attacks, spoofing attacks, etc. IoT attacks comprise different forms of information security attacks that could be targeted on the specific components, network, or data sets. The devices present in the IoT networks may be targeted and physical security attacks may be executed. The majority of the IoT attacks are network-based or are conducted to cause damage to the specific information properties. These are usually deliberate attacks to cause damage to the availability of the IoT application or cause an impact on the confidentiality of the data. Some of the significant threats at the network layer include:

Man-in-the-middle (MITM)attack-

Wireless communications of sensors may be in danger to the man in middle attacks that could threaten the confidentiality, integrity, and availability of the IoT communicant (Neshenko et al., 2019). Wireless attacks could be encryption cracking, rogue wireless devices, Eavesdropping, MAC Spoofing, Packet sniffing, etc. A MITM attack occurs where the attacker without the permission of the authentication user alter the communications between two parties who think that they are secretary communicating with each other. It is just like an eavesdropping attack in which the attacker can insert into the communication of two parties. There is various type of MITM attack which are email hijacking, WIFI eavesdropping, Session Hijacking, DNS spoofing and IP spoofing. For example, an attacker can put in network spyware (a sniffer) on a computer or a server to carry out a spying attack and capture the packet during transmission. Moreover, any device in the network between the transmitting device and the receiving device are vulnerable to attackers, as are the initial and terminal devices themselves.

Denial of service (DoS) attack

A denial of service attack prohibits the regular availability of the services that were being provided in a system. Legitimate users of the system are deprived of the resources. If this attack is launched by numerous malicious nodes, then it is called Distributed Denial of Service (DDoS). A DOS attack will cost time and money to the victim rather than losing information due to service holders switching the services from the original provider concerning the security concerns. DoS attacks can impact network resources, bandwidth, and CPU time (Sherasiya et al., 2016). As the IoT devices and equipment are connected with the internet for 24 h and always on power-on mode, there is a high chance of an attack on the IoT network. Malware payloads can be sent at any time in the home or office IoT network. For instance, 'Mirai' is a botnet that mounted a Distributed Denial of Service (DDoS) attack, which left much of the network unapproachable (Khraisat et al., 2019b).

Distributed denial of service (DDoS)

In a DDoS attack, an attacker briefly compromises several IoT devices into an arrangement known as a botnet and then makes synchronised requests to a server or an array of servers for a specific service, in that way overwhelming the server and make it serve genuine requests from end-users. It usually happens when all the devices are manipulated and messages are overwhelmed by IoT devices and this is mostly used to create a traffic jam in the devices.

Attacks on RPL (routing protocol for low-power and Lossy networks)

Routing Protocol for Low-Power and lossy devices transmitter broadcasts the DODAG Information Object (DIO) during Destination Oriented Directed Acyclic Graph (DODAG) formation. The receiver transmits its updated parent list, sibling list, rang and sends DAO message with route information after receiving the DIO (Mayzaud et al., 2016). After receiving the DIO message, the malicious nodes do not update; rather it always advertises a fake rank. The other non-malicious node receives the DIO message from the malicious node and updates its rank based on the fake rank. After the formation of DODAG, if the node that is transmitting the packet has a malicious node as the preferred parent, transmits the packet to it but the malicious node instead of transmitting the packet to its parent simply drops the packet resulting in zero throughputs.

The low-cost and low-power intensive resource nodes capable of wireless networking allow the viability of new applications such as smart electricity grids as well as mobile health solutions. These power-efficient network devices could be combined with the existing network infrastructure so that they could utilize services already available, including the node's ability to control and data-gathering. A node calculates its rank in the DODAG based on the objective code point specified in a received DIO message. If the node receives multiple DIO messages from its neighbors, then the neighbor that providing the best rank is chosen to be the parent. This way, it forms upwards routes towards the root. The DAO message that contains all possible routable prefixes is sent up the tree to create routes (Khraisat et al., 2019a) downwards. Each node receiving the DAO message aggregates the prefixes and propagates them further upwards, thereby making downwards routes available to parents.

RPL uses three control message types to create and maintain its graph topology and route table. The control messages include the DODAG Information Object (DIO), DODAG Advertisement Object (DAO) and DODAG Information Solicitations (DIS). The creation, maintenance, and discovery of the DODAG topology are done by using DIO. Nodes exchange DODAG messages while the RPL network is being initiated through DIO. The nodes select preferred parents with the help of DIO. RPL uses DAO messages for transmitting the prefix of a node to its ancestor nodes for downward routing purposes. Any unattached node uses the DIS message in the network for soliciting potential parent nodes. When a node cannot obtain DIO, DIS is triggered by a node after a certain time interval. RPL instance is the creation of an RPL network in a DODAG. These RPL instances can have their object functions and can consist of a DODAG. Attacks on RPL topology as the following:

Sybil attack A Sybil attack is defined as a number of nodes faking various peer identities to compromise an IoT ecosystem. It is used to send false data information from a random network. Sybil attacks, where a sensor node claims multiple fake identities, could be highly damaging in the context of an e-health system. Through these attacks, an intruder could use pretend identities to send false information. Consequently, either a real emergency condition is missed. In this attack, a malicious node within a network has multiple identities. A malicious node can affect the routing mechanism, routing protocol and detection algorithm in a peer to peer network.

Selective forwarding attack In a selective forwarding attack, the malicious node acts like a normal node, but it selectively drops some data packets coming from a node or group of nodes (Khan et al., 2012). A malicious node refuses to forward the data coming through it and drops on the way. An infected and malicious node may transmit the message to the wrong path in the network.

Sinkhole attack It is used to attack the traffic of data from the neighborhood nodes. This is mainly carried with the help of a routing algorithm. An internal attack where a malicious node tries to attract the network traffic toward it by advertising fake routing updates is a sinkhole attack. An attack is launched by an attacker by introducing false nodes inside a network (Can & Sahingoz, 2015). The main objective of a sinkhole attack is to misroute the traffic from an area through a compromised node that looks especially attractive to the surrounding nodes (Singh et al., 2015).

Wormhole attack In a wormhole attack, malicious nodes at all times offer an illusion to both the sender device and as well as the receiver device. A virtual tunnel is built up which claims itself the shortest distance between the two ends, which are the malicious nodes so that the base station sends data through it and gets lost on its way. The attacking node captures data and sent it to a distant location from where the data is transmitted locally. The attack can take place either in a hidden mode or a participation mode (Khabbazian et al., 2006).

Hello flooding attack Hello flood attack is one of the most common attacks on the network layer which force IoT devices to send Hello packets to advertise themselves to their neighbors. For connecting the network node broadcast initial message as Hello packet. The Cybercriminal can present himself as neighbor node to numerous nodes by broadcasting Hello message. If a node receives such Hello packet, it will assume that it is inside the radio range of the node that sent that packet.

Blackhole attack In a Blackhole attack, the malicious device incorrectly presents the shortest route to destination and then stealthy drops all packets on its path, making a Blackhole in the network.

Intrusion detection datasets

The evaluation datasets play a vital role in the validation of any IDS approach by allowing us to assess the proposed method's capability in detecting intrusive behaviour. The datasets used for network packet analysis in commercial products are not easily available due to privacy issues. However, there are a few publicly available datasets such as DARPA, KDD, NSL-KDD and ADFA-LD and they are widely used as benchmarks. Existing datasets that are used for building and comparative evaluation of IDS are discussed in this section along with their features and limitations.

DARPA / KDD Cup99

The earliest effort to create an IDS dataset was made by DARPA (Defence Advanced Research Project Agency) in 1998, and they created the KDD98 (Knowledge Discovery and Data Mining (KDD)) dataset. In 1998, DARPA introduced a program at the MIT Lincoln Labs to provide a comprehensive and realistic IDS benchmarking environment (Lincoln Laboratory, 1999). Although this dataset was an essential contribution to the research on IDS, its accuracy and capability to consider real-life conditions have been widely criticized (Creech and Hu, 2014).

These datasets were collected using multiple computers connected to the Internet to model a small US Air Force base of restricted personnel. Network packets and host log files were received. Lincoln Labs built an experimental testbed to obtain 2 months of TCP packets dump for a Local Area Network (LAN), modelling a usual US Air Force LAN. They modelled the LAN as if it were a true Air Force environment, but interlaced it with several simulated intrusions.

The collected network packets were around four gigabytes containing about 4,900,000 records. The test data of 2 weeks had around 2 million connection records, each of which had 41 features and was categorized as normal or abnormal.

The extracted data is a series of TCP sessions starting and ending at well-defined times, between which data flows to and from a source IP address to a target IP address, which contains a large variety of attacks simulated in a military network environment. The 1998 DARPA Dataset was used as the basis to derive the KDD Cup99 dataset, which has been used in Third International Knowledge Discovery and Data Mining Tools Competition (KDD, 1999).

These datasets are out-of-date as they do not contain records of recent malware attacks. For example, attackers' behaviours are different in different network topologies, operating systems, and software, and crime toolkits. Nevertheless, KDD99 remains in use as a benchmark within the IDS research community and is still presently being used by researchers (Alazab et al., 2014; S. Duque and M. N. b. Omar, 2015; Ji et al., 2016).

CAIDA

This dataset contained network traffic traces from Distributed Denial-of-Service (DDoS) attacks and was collected in 2007 (Hick et al., 2007). This type of denial-of-service attack attempts to interrupt regular traffic of a targeted computer, or network by overwhelming the target with a flood of network packets, preventing regular traffic from reaching its legitimate destination computer. One disadvantage of the CAIDA dataset is that it does not contain a diversity of attacks. In addition, the gathered data does not provide features from the whole network, which makes it difficult to distinguish between abnormal and normal traffic flows.

NSL-KDD

NSL-KDD is a public dataset, which has been developed from the earlier KDD cup99 dataset (Tavallaee et al., 2009). A statistical analysis performed on the cup99 dataset raised important issues that heavily influence the intrusion detection accuracy and results in a misleading evaluation of AIDS (Tavallaee et al., 2009).

The main issue in the KDD data set is a large number of duplicate packets. Tavallaee et al. analysed KDD training and test sets and revealed that approximately 78% and 75% of the network packets are duplicated in both the training and testing dataset (Tavallaee et al., 2009). This huge quantity of duplicate instances in the training set would influence machine-learning methods to be biased towards normal instances and thus prevent them from learning irregular instances that are typically more damaging to the computer system. Tavallaee et al. built the NSL-KDD dataset in 2009 from the KDD Cup'99 dataset to resolve the matters stated above by eliminating duplicated records (Tavallaee et al., 2009). The NSL-KDD train dataset consists of 125,973 records and the test dataset contains 22,544 records. The size of the NSL-KDD dataset is sufficient to make it practical to use the whole NSL-KDD dataset without the necessity to sample randomly. This has produced consistent and comparable results from various research works. The NSL_KDD dataset comprises 22 training intrusion attacks and 41 attributes (i.e., features). In this dataset, 21 attributes refer to the connection itself and 19 attributes describe the nature of connections within the same host (Tavallaee et al., 2009).

ISCX 2012

In this dataset, real network traffic traces were analysed to identify normal behaviour for computers from real traffic of HTTP, SMTP, SSH, IMAP, POP3, and FTP protocols (Shiravi et al., 2012). This dataset is based on realistic network traffic, which is labelled and contains diverse attack scenarios.

ADFA-LD and ADFA-WD

Researchers at the Australian Defence Force Academy created two datasets (ADFA-LD and ADFA-WD) as public datasets that represent the structure and methodology of the recent attacks (Creech, 2014). The datasets contain records from both Linux and Windows operating systems; they are created from the evaluation of system-call-based HIDS. Ubuntu Linux version 11.04 was used as the host operating system to build ADFA-LD (Creech and Hu, 2014). Some of the attack instances in ADFA-LD were derived from new zero-day malware, making this dataset suitable for highlighting differences between SIDS and AIDS approaches to intrusion detection. It comprises three dissimilar data categories, each group of data containing raw system call traces. Each training dataset was gathered from the host for normal activities, with user behaviors ranging from web browsing to LATEX document preparation.

ADFA-LD also incorporates system call traces of different types of attacks. The ADFA Windows Dataset (ADFA-WD) provides a contemporary Windows dataset for the evaluation of HIDS. CICIDS 2017.

CICIDS2017 dataset comprises both benign behaviour and also details of new malware attacks: such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS (Sharafaldin et al., 2018). This dataset is labelled based on the timestamp, source and destination IPs, source and destination ports, protocols and attacks. Complete network topology was configured to collect this dataset which contains Modem, Firewall, Switches, Routers, and nodes with different operating systems (Microsoft Windows (like Windows 10, Windows 8, Windows 7, and Windows XP), Apple's macOS iOS, and open-source operating system Linux). This dataset contains 80 network flow features from the captured network traffic.

IoT botnet

The Bot-IoT dataset, which includes normal IoT network traffic along with a variety of attacks, is used to evaluate our proposed framework. This Dataset is selected as it represents a realistic IoT ecosystem environment. The dataset contains DDoS, DoS, OS and Service Scan, Keylogging and Data exfiltration attacks.

All this data is pre-processed to identify network-level patterns for diverse kinds of traffic that devices create, and use these patterns to detect any intrusion behaviours in the IoT Infrastructure (Koroniotis et al., 2018).

Comparison of public IDS datasets

Since machine learning techniques are applied in AIDS, the datasets that are used for the machine learning techniques are very important to assess these techniques for realistic evaluation. Table 10 summarizes the characteristics of the datasets. We found that the well-known KDD'99 or similar sets crafted for a wired network environment will not lead to the creation of optimized IDS targeting the IoT ecosystem.

Challenges of IoT IDS

In the Internet of Things (IoT) era, the large communicated devices are quickly rising. The security of communications in the IoT environment, using the previously IDSs raise challenges and prospects for future research work.

No doubt that there has been a lot of research in the area of IDSs, still there are many important matters to work on. IDSs have to be more accurate, with the capability to detect a varied range of intrusions with fewer false alarms and other challenges.

Feature – engineer extraction

The detection effect of this method is highly dependent on the design of the traffic features used in training. The IDS accuracy performs often different when various feature sets of network traffic are used. No standard research direction currently exists for the directing of a feature set that precisely differentiate network traffic (Wang et al., 2018).

IoT device limitations

IoT devices have small memory space, which is a challenge to keep track of as the system runs continuously and can be overwritten due to low memory storage, contributing to the possibility that important evidence is missing. For example, limited memory space in IoT devices, data could be easily overwritten or some IoT devices do not store data. There may be a way to save the data by transferring the data to the storage device, but this choice is not always effective because data can be easily altered during transfer to the local storage device. The other IoT device's limitation is computing power. A cybercriminal could consume the stored energy by producing a flood of legitimate or malicious messages, expose the sensors unavailable for legitimate users (Neshenko et al., 2019).

In (Granjal et al., 2015), the authors investigated the difficulties of IoT intrusion detection systems at the network layer. The research in (Xiao et al., 2018) overviewed a discussion of the ML technique's relevance in the context of IoT intrusion detection systems. Furthermore, they recognised limited bandwidth, computation power and lack of sufficient memory as bottlenecks in any implementation of intrusion detection system based on the machine learning for IoT networks.

Since some IoT devices are conveyed in situations where charging isn't accessible, they just have a constrained vitality to execute the designer of IDS and heavy IDS analysis can drain the devices' resources. This is required to design a Lightweight Intrusion Detection System for the Internet of Things that uses the least possible security necessities on the IoT device. A lightweight IDS system could be designed by reducing the complex features extraction and features. A limited number of features should be extracted from raw data to achieve accuracy in detecting an intrusion in the IoT ecosystem. Feature selection is helpful to decrease the computational difficulty, eliminate data redundancy, enhance the detection rate of the machine learning techniques, simplify data, and reduce false alarms. In this line of research, some methods have been applied to develop a lightweight IoT IDSs.

Problems of smart devices

If there is an IoT device is poorly configured or slow to release firmware updates for smart devices, it might cause some security issues. For example, IoT devices

Table 10 The compassion of datasets (✓ = True, ✖ = False)

Dataset	Real Traffic	Label data	IoT traces	Zero-day attacks	Full packet captured	Year
DARPA 98	✓	✓	✖	✖	✓	1998
KDDCUP 99	✓	✓	✖	✖	✓	1999
CAIDA	✓	✖	✖	✖	✖	2007
NSL-KDD	✓	✓	✖	✖	✓	2009
ISCX 2012	✓	✓	✖	✖	✓	2012
ADFA-WD	✓	✓	✖	✓	✓	2014
ADFA-LD	✓	✓	✖	✓	✓	2014
CICIDS2017	✓	✓	✖	✓	✓	2017
Bot-IoT	✓	✓	✓	✓	✓	2018

could be used to perform criminal activities, or an attacker who has gained access to an IoT device could spy. Another problem is preconfigured passwords set by the manufacturer. For example, the authentication login can simply be found on the Internet. One more issue that makes the cybercriminal's activities easier is that various IoT devices have their communication ports open to the external network.

Overhead traffic

In traditional network environments, traffic-based trust computation performs well in detecting insider attacks. However, with the high-speed network connection, huge packets have emerged as a challenge because the traffic might critically go over the limited processing ability of an IDS.

Overhead traffic can make an IDS drop numerous parcels without appropriate checking, reducing the security level of its whole computer system. In the era of IoT, network packets are more dynamic and considerably more difficult, making the challenge even more challenging.

Heterogeneity device type

Heterogeneous means are diverse. The IoT links various types of devices so that the physical and virtual world can communicate. It is possible to connect items and objects in general such as smartphones, smartwatches, refrigerators, air conditioners, sensors, automated home systems, automotive systems, robots, tablets and mobile devices everywhere.

The main challenge is to link all devices to each other is that the various heterogeneous devices are running on various platforms and frameworks. The IoT features, the mass of diverse devices, complexity at the network level, various communication protocols communicate including an ultra-largescale network of things, device, and network-level heterogeneity, and huge amounts of actions produced naturally by these sensors will make the development of the IDS a very challenging task.

At this time, there is a deficiency of commonly standardized IoT IDS that deals with the heterogeneity of underlining communication technologies and provides a transparent naming service to various applications. Until now, it still a challenge for IDS to detect different attacks among a number of IoT devices because of the devices' large quantity and dynamic nature.

Privacy

The majority of IoT datasets are existing with big organizations that are unwilling to share it so certainly. Access to copyrighted datasets or privacy considerations. These are more general in the area with personal data such as healthcare and education.

Feature extraction

Feature extraction is the task of getting the network traffic from IoT devices' communication. In IDS, it is essential to extract features that capture the context and purpose of each packet crossing the network. For instance, the packet may be a normal connection to communicate with a server, or it might be one of the billions of malicious packets transmitted in a purpose to source malicious activities (Mirsky et al., 2018). The challenge with extracting these types of features from IoT network traffic is that packets from different subnet networks are overlapped, there could be various networks connection at any given instant, and a high-speed connection (S. P. R. M, 2020).

Big IoT data

Increase in volume, variety, and velocity of IoT device data and a rapidly growing number of connected devices. Scalability issues often arise as more and more physical things are communicated to the network (Tang et al., 2019). When the number of things is large, scalability is challenging at different levels, including data transfer and networking, data processing and management, and service provisioning. Big volumes of data transmission across the IoT ecosystem at the same time can also produce regular delays, conflict, and communication matters. It is a challenging task to develop networking technologies and standards that can allow data gathered by a large number of devices to move efficiently within IoT networks.

Immaturity of communication protocol

IDS is usually incorporated with IoT protocol to detect IoT attacks. The immaturity of security protocol is impacted by developing stabilized IDS. Hence, IDS extracted the features from the network protocols (Neshenko et al., 2019). The variety of IoT devices and IoT protocols are challenges that are certainly valuable of being followed for developing IoT IDS and resiliency. IoT ecosystem is run by wireless networking protocols operating at physical and data link layers as well as some different protocols and standards that are primarily designed for IoT applications. Wireless personal area networks (WPANs) standards are employed for short-range communication such as Bluetooth and ZigBee. Another short-range wireless communication protocols used by various IoT sensors are Near-Field Communication (NFC). Hybrid standards such as IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) offer low-power IoT objects with smaller encapsulation by compressing the header (Benkhelifa et al., 2018). For a longer range, cellular networks are mostly employed.

Data collection

Data collection from every IoT sensor is a challenge. During the data can be updated or altered or the Data can be updated or altered or Data can be vanished. It is also challenging because Unknown or not accessible physical location, locating evidence in large and changing systems, decentralized data, and the data is erased as soon as IoT gets rebooted. Another data collection challenges that cloud service providers do not disclose any information regarding the cloud's internal structure to protect the consumer data. For example, the data collected may be in a different format from IoT devices than the data preserved in the cloud as it can be encrypted before saving the data in the cloud. Also, restorative information of the tolerance gathered by the specialist organizations has comparative difficulties. In addition, a cyber-security expert is needed for collecting a dataset containing both normal traffic and network attacks.

Unavailability of training datasets

Effective utilization of machine learning and Deep learning needs significant datasets that are at present missing. Moreover, the principles and arrangements required for characterizing the learning techniques despite everything should be investigated. Also, true datasets from the genuine physical condition are required to break down and analyse the presentation of different DL and RL calculations. Until this point in time, endeavours have been made to adapt to this test. However, more research is required right now in this direction.

The most popular public datasets used for IDS research have been explored and their data collection techniques, evaluation results, and limitations have been discussed. There is a requirement for newer and more comprehensive datasets that consist of a broad spectrum of malware activities because the normal activities are changing frequently and might not remain effective over time. A new malware dataset is required, as most of the existing machine learning techniques are trained and evaluated on the knowledge provided by the old dataset such as DARPA/ KDD99, which do not include newer malware activities. Therefore, testing is done using these datasets collected in 1999 only, because they are publicly available and no other alternative and acceptable datasets are available. While widely accepted as benchmarks, these datasets no longer represent contemporary zero-day attacks and the IoT ecosystem (Venkatraman & Alazab, 2018). Though the ADFA dataset contains many new attacks, it is not adequate. For that reason, testing of AIDS using these datasets does not offer a real evaluation and could result in inaccurate claims for their effectiveness.

Challenges of IoT IDS for ICS

A varied variety of industrial IoT systems have been used in recent years such as transportation, manufacturing, retailing and smart city infrastructures. With the developments in wireless communication, smartphone, healthcare (e.g. remote patient 24-h care o), smart grid, home automation (e.g. security, heating and lighting control) and smart cities (e.g. distributed pollution monitoring, smart lightning systems), and sensor network technologies, more and more connected things are being used in IoT. Cyber-Physical Systems (CPS) relies on the IoT ecosystem in that they are united of both physical sensors and actuators networked with computer-based control systems. Industrial Control Systems (ICSs) are commonly comprised of two components: Supervisory Control and Data Acquisition (SCADA) hardware which receives information from sensors and then controls the mechanical machines; and the software that enables human administrators to control the machines.

Cyber-attacks on ICSs are a great challenge for the IDS due to the unique architectures of ICSs as the attackers are currently focusing on ICSs. A standout amongst the recent attacks against ICSs is the Stuxnet attack, which is known as the first cyber-warfare weapon. Dissimilar to a typical attack, the primary target of Stuxnet was probably the Iranian atomic program (Nourian & Madnick, 2018). Attacks that could target ICSs could be state-sponsored, or they might be launched by competitors, internal attackers with a malicious target, or even hacktivists.

The potential consequences of compromised ICS can be devastating to public health and safety, national security, and the economy. Compromised ICS systems have led to extensive cascading power outages, dangerous toxic chemical releases, and explosions. For reliable, safe, and flexible performance, it is required to use secure ICSs.

It is critical to have IDS for ICSs that take into account unique architecture, real-time operation and dynamic environment to protect the facilities from the attacks. Some critical attacks on ICSs are given below:

- In 2008, Conficker malware-infected ICS systems, such as an airplane's internal systems. Conficker disables many security features and automatic backup settings, erases stored data and opens associations to get commands from a remote PC (Pretorius & van Niekerk, 2016).
- In 2009, a 14-year-old schoolboy hacked the city's tram system and used a homemade remote device to redirect many trams, injuring 12 passengers (Rege-Patwardhan, 2009).

- In 2017, WannaCry ransomware spread globally and seriously affected the National Health System, UK and prevented emergency clinic specialists from using health systems (Mohurle & Patil, 2017).

Since Microsoft no longer creates security patches for legacy systems, they can simply be attacked by new types of ransomware and zero-day malware.

Similarly, it may not be possible to fix or update the operating systems of ICSs for legacy applications.

A robust IDS is a solution for protecting industries from the threat of cyber-attacks. The current IDS techniques, as proposed in the literature focus are at the software level. A newer detection method is required for detecting the zero-day and complex attacks at the software level without having any prior knowledge. This could be implemented by combining both hardware and software intrusion detection systems and extracting useful features.

Challenge of IoT IDS on intrusion evasion detection

The main challenge for SIDS and AIDS is to detect attacks masked by evasion techniques. The ability of evasion techniques would be determined by the ability of IDS to bring back the original signature of the attacks or create new signatures to cover the modification of the attacks. The robustness of IDS to various evasion techniques still needs further investigation. For example, SIDS in regular expressions can detect the deviations from simple mutations such as manipulating space characters, but they are still useless against several obfuscation techniques usually used by hackers to conceal malware such as encryption and packing.

Discussion and conclusion

In this paper, we have presented, in detail, a critical review of IoT intrusion detection system methodologies, deployment strategy, validation strategy, Dataset and technologies with their advantages and limitations. Several intrusion detection systems have been proposed to detect IoT attacks are reviewed. However, such approaches may have the problem of detecting all IoT attacks due to IoT architecture. We summarized the results of recent research and explored the contemporary models on the performance improvement of IoT IDS as a solution to overcome IoT security issues. We have also shed light on the restrictions of the customary IoT Intrusion detection system. Then we discussed the existing IDS and presented the challenges and future research directions.

To develop reliable IoT IDS based on heterogeneous device categories, a novel IDS must be developed. We recognize four elements that have a vital feature in the building of reliable IDS for the IoT. First, be low on false alarms due to the large volume of data. Second, be highly adaptive to extreme IoT communication systems due to unexpected behavior in IoT sensors that once appeared usual may start considering attacks. Third, be able to detect zero-day attacks as new vulnerabilities are exposed. Fourth, be autonomous IDS and use contemporary techniques of machine learning and deep learning that can learn from the big IoT data. In addition, Future IoT IDS should have features including self-configuration, self-optimization, self-protection, and self-healing.

In conclusion, we believe this review may provide an important contribution to the security researchers, by reviewing the contemporary status of this significant and very dynamic area of research, facilitating researcher interested in developing novel IDS to address IoT security in the context of communication for the IoT.

Acknowledgments
The research is supported by the Internet Commerce Security Laboratory, Federation University Australia. The authors are grateful to the Centre for. Informatics and Applied Optimization (CIAO) for their support.

Authors' contributions
Ansam and Ammar have participated presented, in detail, a survey of intrusion detection system methodologies, types, and technologies with their advantages and limitations. Several machine learning techniques have been proposed to detect zero-day attacks are reviewed. Ammar, Michael and Mamoun has gone through the article and add IoT attacks taxonomy. The author(s) read and approved the final manuscript.

Funding
This work was carried out within the Internet Commerce Security Lab, which is funded by Westpac Banking Corporation.

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 18 November 2020 Accepted: 4 February 2021
Published online: 08 March 2021

References

A. Abbasi, J. Wetzels, W. Bokslag, E. Zambon, S. Etalle, "On emulation-based network intrusion detection systems," in Research in attacks, Intrusions and Defenses: 17th International Symposium, RAID 2014, Gothenburg, Sweden, September 17–19, 2014. Proceedings, A. Stavrou, H. Bos, G. Portokalidis, Cham: Springer International Publishing, 2014, pp. 384–404

Aburomman AA, Ibne Reaz MB (2016) A novel SVM-kNN-PSO ensemble method for intrusion detection system. Appl Soft Comput 38:360–372

Aburomman AA, Reaz MBI (2017) A survey of intrusion detection systems based on ensemble and hybrid classifiers. Comput Security 65:135–152

Agrawal S, Agrawal J (2015) Survey on anomaly detection using data mining techniques. Procedia Computer Science 60:708–713

Alazab A, Hobbs M, Abawajy J, Alazab M (2012) Using feature selection for intrusion detection system. In: 2012 International Symposium on Communications and Information Technologies (ISCIT), pp 296–301

Alazab A, Hobbs M, Abawajy J, Khraisat A, Alazab M (2014) Using response action with intelligent intrusion detection and prevention system against web application malware. Inf Manag Comput Secur 22(5):431–449

Alcaraz C (2018) Cloud-assisted dynamic resilience for cyber-physical control systems. IEEE Wirel Commun 25(1):76–82

Annachhatre C, Austin TH, Stamp M (2015) Hidden Markov models for malware classification. J Comput Virol Hack Technique 11(2):59–73

Axelsson S (2000) "Intrusion detection systems: A survey and taxonomy," Technical report

Bajaj K, Arora A (2013) Dimension reduction in intrusion detection features using discriminative machine learning approach. IJCSI Int J Comput Sci Issues 10(4): 324–328

Benkhelifa E, Welsh T, Hamouda W (2018) A critical review of practices and challenges in intrusion detection systems for IoT: toward universal and resilient systems. IEEE Commun Survey Tutor 20(4):3496–3509

Bhuyan MH, Bhattacharyya DK, Kalita JK (2014) Network anomaly detection: methods, systems and tools. IEEE Commun Survey Tutorial 16(1):303–336

Breiman L (1996) Bagging predictors. Machine Learn 24(2):123–140

Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surveys Tutorial 18(2):1153–1176

Butun I, Morgera SD, Sankar R (2014) A survey of intrusion detection systems in wireless sensor networks. IEEE Commun Survey Tutorial 16(1):266–282

Can O, Sahingoz OK (2015) A survey of intrusion detection systems in wireless sensor networks. In: 2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), pp 1–6 IEEE

Cervantes C, Poplade D, Nogueira M, Santos A (2015) Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp 606–611 IEEE

Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P (2019) Network Intrusion Detection for IoT Security Based on Learning Techniques, in IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2671–2701, thirdquarter 2019. https://doi.org/10.1109/COMST.2019.2896380

Chao L, Wen S, Fong C (2015) CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge Based Syst 78:13–21

Chebrolu S, Abraham A, Thomas JP (2005) Feature deduction and ensemble design of intrusion detection systems. Comput Security 24(4):295–307

Cho EJ, Kim JH, Hong CS (2009) Attack model and detection scheme for botnet on 6LoWPAN. Springer Berlin Heidelberg, Berlin, pp 515–518

Creech and Hu (2014) A semantic approach to host-based intrusion detection systems using contiguous and Discontiguous system call patterns. IEEE Trans Comput 63(4):807–819

Creech G (2014) Developing a high-accuracy cross platform host-based intrusion detection system capable of reliably detecting zero-day attacks. University of New South Wales, Canberra

da Costa KAP, Papa JP, Lisboa CO, Munoz R, de Albuquerque VHC (2019) Internet of Things: A survey on machine learning-based intrusion detection approaches. Comput Network 151:147–157

Debar H, Dacier M, Wespi A (2000) A revised taxonomy for intrusion-detection systems. Annales des télécommunications 55(7–8):361–378 Springer

Diro AA, Chilamkurti N (2018) Distributed attack detection scheme using deep learning approach for internet of things. Futur Gener Comput Syst 82: 761–768

Dua S, Du X (2016) Data Mining and Machine Learning in Cybersecurity Publishers Auerbach. Publications Location UK

Elhag S, Fernández A, Bawakid A, Alshomrani S, Herrera F (2015) On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems. Expert Syst Appl 42(1):193–202

Granjal J, Monteiro E, Silva JS (2015) Security for the internet of things: a survey of existing protocols and open research issues. IEEE Commun Survey Tutor 17(3):1294–1312

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD explorations newsletter 11(1):10–18

Hick P, Aben E, Claffy K, Polterock J (2007) "the CAIDA DDoS attack 2007 dataset," ed

H. Hindy et al., "A taxonomy and survey of intrusion detection system design techniques, network threats and datasets," *arXiv preprint arXiv: 1806.03517,* 2018

Hodo E et al (2016) Threat analysis of IoT networks using artificial neural network intrusion detection system. In: 2016 International Symposium on Networks, Computers and Communications (ISNCC), pp 1–6

Hoque MAM, Bikas AN (2012) An implementation of intrusion detection system using genetic algorithm. arXiv preprint arXiv:1204.1336. Chicago; 109–120

Jabbar MA, Aluvalu R, S. S. Reddy S (2017) RFAODE: A Novel Ensemble Intrusion Detection System. Procedia Comput Sci 115:226–234

Ji S-Y, Jeong B-K, Choi S, Jeong DH (2016) A multi-level intrusion detection method for abnormal network behaviors. J Network Comput Application 62(Supplement C):9–17

KDD. (1999). The 1999 KDD intrusion detection. Available: http://kdd.ics.uci.edu/databases/kddcup99/task.html

Kenkre PS, Pai A, Colaco L (2015) Real time intrusion detection and prevention system. In: Satapathy SC, Biswal BN, Udgata SK, Mandal JK (eds) Proceedings of the 3rd international conference on Frontiers of intelligent computing: theory and applications (FICTA) 2014: volume 1. Springer International Publishing, Cham, pp 405–411

Khabbazian M, Mercier H, Bhargava VK (2006) Nis02–1: Wormhole attack in wireless ad hoc networks: Analysis and countermeasure. In: IEEE Globecom 2006, pp 1–6 IEEE

Khan WZ, Xiang Y, Aalsalem MY, Arshad Q (2012) The selective forwarding attack in sensor networks: Detections and countermeasures. Int J Wireless Microwave Technol (IJWMT) 2(2):33

Khraisat A, Gondal I, Vamplew P (2018) An Anomaly Intrusion Detection System Using C5 Decision Tree Classifier. In: Trends and Applications in Knowledge Discovery and Data Mining, Cham. Springer International Publishing, pp 149–155

Khraisat A, Gondal I, Vamplew P, Kamruzzaman J (2019a) "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*. J Article 2(1):20

Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Alazab A (2019b) A Novel Ensemble of Hybrid Intrusion Detection System for Detecting Internet of Things Attacks. Electronics 8(11):1210

Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Alazab A (2020) Hybrid Intrusion Detection System Based on the Stacking Ensemble of C5 Decision Tree Classifier and One Class Support Vector Machine. Electronics 9(1):173

Koc L, Mazzuchi TA, Sarkani S (2012) A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. Expert Syst Appl 39(18):13492–13500

Koroniotis N, Moustafa N, Sitnikova E, Turnbull B (2018) "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset," arXiv preprint arXiv:1811.00701

Kreibich C, Crowcroft J (2004) Honeycomb: creating intrusion detection signatures using honeypots. SIGCOMM Comput Commun Rev 34(1):51–56

Li Y, Xia J, Zhang S, Yan J, Ai X, Dai K (2012) An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Syst Appl 39(1):424–430

Liao H-J, Lin C-HR, Lin Y-C, Tung K-Y (2013b) Intrusion detection system: a comprehensive review. J Netw Comput Appl 36(1):16–24

Liao H-J, Richard Lin C-H, Lin Y-C, Tung K-Y (2013a) Intrusion detection system: A comprehensive review. J Network Comput Appl 36(1):16–24

Lin C, Lin Y-D, Lai Y-C (2011) A hybrid algorithm of backward hashing and automaton tracking for virus scanning. IEEE Trans Comput 60(4):594–601

Lin W-C, Ke S-W, Tsai C-F (2015) CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge Based Syst 78(Supplement C):13–21

MIT Lincoln Laboratory. (1999). DARPA Intrusion Detection Data Sets. Available: https://www.ll.mit.edu/ideval/data/

Lunt TF (1988) Automated audit trail analysis and intrusion detection: a survey. In: Proceedings of the 11th National Computer Security Conference, Washington, D.C.: National Bureau of Standards, National Computer Security Center; vol 353, Baltimore

Mayzaud A, Badonnel R, Chrisment I (2016) A taxonomy of attacks in RPL-based internet of things. Int J Network Security 18(3):459–473

McHugh J (2000) Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Trans Inf Syst Secur 3(4):262–294

Meiners CR, Patel J, Norige E, Torng E, Liu AX (2010) "Fast regular expression matching using small TCAMs for network intrusion detection and prevention systems," presented at the proceedings of the 19th USENIX conference on security, Washington, DC

Meshram A, Haas C (2017) Anomaly Detection in Industrial Networks using Machine Learning: A Roadmap. In: Beyerer J, Niggemann O, Kühnert C (eds) Machine Learning for Cyber Physical Systems: Selected papers from the International Conference ML4CPS 2016. Springer Berlin Heidelberg, Berlin, pp 65–72

Mirsky Y, Doitshman T, Elovici Y, Shabtai A (2018) "Kitsune: an ensemble of autoencoders for online network intrusion detection," arXiv preprint arXiv: 1802.09089

Modi C, Patel D, Borisaniya B, Patel H, Patel A, Rajarajan M (2013) A survey of intrusion detection techniques in Cloud. J Network Comput Appl 36(1):42–57

Mohurle S, Patil M (2017) A brief study of wannacry threat: Ransomware attack 2017. Int J Adv Res Comput Sci 8(5):1938–1940

Moustafa N, Turnbull B,Choo KR (2019) "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," IEEE Internet of Things Journal, vol. 6, pp. 4815-4830

Murray SN, Walsh BP, Kelliher D, O'Sullivan DTJ (2014) Multi-variable optimization of thermal energy efficiency retrofitting of buildings using static modelling and genetic algorithms – A case study. Build Environ 75(Supplement C):98–107

Neshenko N, Bou-Harb E, Crichigno J, Kaddoum G, Ghani N (2019) Demystifying IoT security: an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations. IEEE Commun Survey Tutorial 21(3): 2702–2733

Nourian A, Madnick S (2018) A systems theoretic approach to the security threats in cyber physical systems applied to Stuxnet. IEEE Transact Dependable Secure Comput 15(1):2–13

Pretorius B, van Niekerk B (2016) Cyber-security for ICS/SCADA: a south African perspective. Int J Cyber Warfare Terrorism (IJCWT) 6(3):1–16

Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106

Quinlan JR (2014) C4. 5: Programs for Machine Learning; Morgan Kaufmann Publishers Inc.: San Francisco; 2014;8

Rathore S, Park JH (2018) Semi-supervised learning based distributed attack detection framework for IoT. Appl Soft Comput 72:79–89

Rege-Patwardhan A (2009) Cybercrimes against critical infrastructures: a study of online criminal organization and techniques. Crim Justice Stud 22(3): 261–271

K. Riesen, H. Bunke, "IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning," in Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, SSPR & SPR 2008, Orlando, USA, December 4–6, 2008. Proceedings, N. da Vitoria Lobo et al., Berlin: Springer Berlin Heidelberg, 2008, pp. 287–297

Roesch M (1999) Snort-lightweight intrusion detection for networks. In: Proceedings of LISA '99: 13th Systems Administration Conference Seattle, Seattle, pp 229–238

Rutkowski L, Jaworski M, Pietruczuk L, Duda P (2014) Decision trees for mining data streams based on the Gaussian approximation. IEEE Trans Knowl Data Eng 26(1):108–119

S. Duque and M. N. b. Omar (2015) Using Data Mining Algorithms for Developing a Model for Intrusion Detection System (IDS). Procedia Comput Sci 61(Supplement C):46–51

S. P. R. M et al (2020) An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. Comput Commun 160: 139–149

Sharafaldin I, Lashkari AH, Ghorbani AA (2018) Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In: ICISSP, pp 108–116

Shen C, Liu C, Tan H, Wang Z, Xu D, Su X (2018) Hybrid-augmented device fingerprinting for intrusion detection in industrial control system networks. IEEE Wirel Commun 25(6):26–31

Sherasiya T, Upadhyay H, Patel HB (2016) A survey: Intrusion detection system for internet of things. Int J Comput Sci Eng (IJCSE) 5(2):91–98

Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA (2012) Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput Security 31(3):357–374

Singh AP, Singh P, Kumar R (2015) A Review on Impact of Sinkhole Attack in Wireless Sensor Networks. Int J 5(8)

Subramanian S, Srinivasan VB, Ramasa C (2012) Study on classification algorithms for network intrusion systems. J Commun Comput 9(11):1242–1246

Symantec (2017) Internet Security Threat Report 2017, vol 22

Tang M, Alazab M, Luo Y (2019) Big data for Cybersecurity: vulnerability disclosure trends and dependencies. IEEE Transact Big Data 5(3):317–329

Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp 1–6

Thaseen S, Kumar CA (2013) An analysis of supervised tree based classifiers for intrusion detection system. In: 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, pp 294–299

Vasan D, Alazab M, Venkatraman S, Akram J, Qin Z (2020a) MTHAEL: cross-architecture IoT malware detection based on neural network advanced ensemble learning. IEEE Trans Comput 69(11):1654–1667

Vasan D, Alazab M, Wassan S, Naeem H, Safaei B, Zheng Q (2020b) IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. Computer Networks 171:107138

Vasan D, Alazab M, Wassan S, Safaei B, Zheng Q (2020c) Image-Based malware classification using ensemble of CNN architectures (IMCEC). Comput Security 92:101748

Venkatraman S, Alazab M (2018) Use of Data Visualisation for Zero-Day Malware Detection. Security Commun Network 2018:1728303

Vigna G, Kemmerer RA (1999) NetSTAT: a network-based intrusion detection system. J Comput Secur 7:37–72

Wang G, Hao J, Ma J, Huang L (2010) A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. Expert Syst Application 37(9):6225–6232

Wang W et al (2018) HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access 6: 1792–1806

Wang X, Han Y, Leung VC, Niyato D, Yan X, Chen X (2020) Convergence of edge computing and deep learning: a comprehensive survey. IEEE Commun Survey Tutorial

Xiao L, Wan X, Lu X, Zhang Y, Wu D (2018) IoT security techniques based on machine learning: how do IoT devices use AI to enhance security? IEEE Signal Process Mag 35(5):41–49

Yang X, Tian YL (2012) EigenJoints-based action recognition using Naïve-Bayes-Nearest-Neighbor. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp 14–19

Yang Y, Wu L, Yin G, Li L, Zhao H (2017) A survey on security and privacy issues in internet-of-things. IEEE Internet Things J 4(5):1250–1258

Yar M, Steinmetz KF (2019) Cybercrime and society. SAGE Publications Limited

Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5:21954–21961

Zarpelao BB, Miani RS, Kawakani CT, de Alvarenga SC (2017) A survey of intrusion detection in internet of things. J Netw Comput Appl 84:25–37

Zhang G, Yan C, Ji X, Zhang T, Zhang T, Xu W (2017) Dolphinattack: Inaudible voice commands. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp 103–117

Publisher's Note

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