symmetry

MDPI

Review

An Intrusion Detection System for the Internet of Things Based on Machine Learning: Review and Challenges

Ahmed Adnan [1,*], Abdullah Muhammed [1,*], Abdul Azim Abd Ghani [2], Azizol Abdullah [1] and Fahrul Hakim [1]

[1] Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, University Putra Malaysia, Serdang 43300, Malaysia; azizol@upm.edu.my (A.A.); fahrul@upm.edu.my (F.H.)
[2] Department of Software Engineering and Information System, Faculty of Computer Science and Information Technology, University Putra Malaysia, Serdang 43300, Malaysia; azim@upm.edu.my
* Correspondence: gs49566@student.upm.edu.my (A.A.); abdullah@upm.edu.my (A.M.)

Abstract: An intrusion detection system (IDS) is an active research topic and is regarded as one of the important applications of machine learning. An IDS is a classifier that predicts the class of input records associated with certain types of attacks. In this article, we present a review of IDSs from the perspective of machine learning. We present the three main challenges of an IDS, in general, and of an IDS for the Internet of Things (IoT), in particular, namely concept drift, high dimensionality, and computational complexity. Studies on solving each challenge and the direction of ongoing research are addressed. In addition, in this paper, we dedicate a separate section for presenting datasets of an IDS. In particular, three main datasets, namely KDD99, NSL, and Kyoto, are presented. This article concludes that three elements of concept drift, high-dimensional awareness, and computational awareness that are symmetric in their effect and need to be addressed in the neural network (NN)-based model for an IDS in the IoT.

Keywords: intrusion detection system; concept drift; high dimensionality; computational complexity

check for updates

Citation: Adnan, A.; Muhammed, A.; Abd Ghani, A.A.; Abdullah, A.; Hakim, F. An Intrusion Detection System for the Internet of Things Based on Machine Learning: Review and Challenges. *Symmetry* **2021**, *13*, 1011. https://doi.org/10.3390/sym13061011

Academic Editor: Jan Awrejcewicz

Received: 26 February 2021
Accepted: 20 March 2021
Published: 4 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The security of technology is a continuously developing and emerging topic. More advancements of technologies lead to more vulnerability and threat of attacks. Traditional methods of security are not valid, since more intelligent attacks are being launched. This has led researchers to exploit another aspect for protecting systems from attacks, which is data that is generated from almost every device. It is well known, that recent technologies generate massive data from a wide range of sources, for example, smartphones which provide a source of multistream data from their sensor sets such as accelerometers, gyros, and global positioning system [1]. It also includes the Internet of Things (IoT), which has supported the emergence of new concepts of stream data generation such as health care IoT [2].

Moreover, industrial IoT [3] requires big data analysis for management of the industrial process as well as for developing supervisory control and data acquisition based on IoT data (SCADA-IoT) [4]. In addition, IoT applications in smart cities such as disseminating smart sensors' codes by using vehicles [5] and public transportation [6] to help automate railways, roadways, airways, and marine, and therefore enhance customers' experiences regarding the way goods are transported, tracked, and delivered. Last but not least, sensors in solar panels are updated based on weather conditions in the IoT to enable solar energy outlets [7]. These tremendous amounts of continuously generated data have opened the door to numerous applications by analyzing the data and identifying hidden patterns and triggering needed actions accordingly.

In the last few years, the world has witnessed an impressive revolution of artificial intelligence (AI) and its applications in various sectors [8–10]. This has enabled the creation

of real working systems for many tasks that were considered to be a type of fiction in the past couple of decades. Today, it is possible to develop and implement an AI system that identifies people in airport and public locations and links their activities with their entire life's recorded data. Furthermore, AI-enabled security systems provide the ability to identify human activities and trigger possible alarms in the case of suspicious behavior. Such tasks are feasible due to the rapid development of hardware computational power and low cost of sensing technologies, and the emergence of advanced internet protocols, including the IoT. These complicated artificial tasks operate based on complex system architecture from the perspective of communication, data processing, and algorithms; however, the complex architecture makes the system vulnerable to attacks. This has motivated researchers to extend any advanced system with a security layer or framework that identifies the attack and provides an alarm when needed to enable required action.

Exploiting AI for developing various models that can analyze network data instantly and predict its nature is the core of security systems. This high velocity and massive amount of IoT-generated data have created new machine learning problems in data science. Those problems have been defined in various topics such as clustering, classification, forecasting, and regression [11]. Almost every year, at least one or multi attacks are launched, causing failure of various cloud-based platforms and applications or causing a threat of data leakage. The streaming nature of operating attacks make them a good example of stream data system applications. Furthermore, any attack detection/identification system has to analyze the data, extract its features, and interpret it explicitly or implicitly using machine learning. Here, machines are trained on previous experience based on labeled data from famous and rare attacks. This implies the stream data learning nature of attack detection systems.

IoT systems consisting of things, services, and networks are vulnerable to network attacks, physical attacks, software attacks, and privacy leakage [12]. For example, in the industrial IoT, it has been observed that many industrial protocols have not been immune to attacks such as Modbus, BACnet, DNP3, and MQTT [4], and therefore new machine learning-related applications in the IoT called attack prediction and identification have been applied [12,13]. Another example is increasing the IoT immunity by preserving the outsourced data's integrity through various security technologies such as cryptosystem and blockchain [14].

The stream data analysis has had an evolving nature that makes any learning model subject to failure in many evolving scenarios. This becomes more severe when the application of the learning and predicting algorithm attacks the prediction of the network. Hence, nowadays, one emerging research area is concept drift-free attack prediction systems [4] based on the drift concept, i.e., changing the distribution of various classes of the data. To countering this issue, better-performing prediction algorithms need to be produced. The data in the concept drift scenarios is non-stationary; the statistical descriptors of data are highly dynamic, making the prediction poor without any continuous update of the algorithm's internal parameters.

The problem of sequential learning in the IoT is considered to be a high-dimensional problem [15], due to the massive amount of generated data. This has also motivated researchers to develop algorithms capable of handling massive data analysis and providing predictions efficiently without consuming huge resources of computational power and memory. For instance, installing the sensor at each junction and road in an evenly distributed manner is expected for smart transportation in a city. Each sensor provides raw data for certain variables such as vehicle counting and speed, location. Next, all of these variables are gathered and provided to the cloud for analysis and decision making, such as vehicle routing or traffic signal control. Any attack or data manipulation can lead to false decision making, causing considerable economic damage. Thus, it is crucial to possess good performing prediction algorithms to learn sequentially with the fewest false predictions.

There is a real-time constraint in many sequential learning applications, in which it important to have light computational algorithms, an additional challenge because of

the multi-variant aspect of the problem. Hence, most developed systems for stream data learning and prediction have a careful computational design, which meets the real-time constraint that changes according to the application [16].

In this study, we aim to address the various stream data learning problems in general, as well as stream data-based attack detection and identification in the IoT. Hence, the specific challenges of stream data learning for attack detection and identification in the IoT is addressed. They are summarized under three main challenges: vulnerability to concept drift, high dimensionality data issues, and the issue of real-time constraint (hard or soft) according to the application.

An intrusion detection system (IDS) is an important security topic with high association with firms' legal, reputation, and economic concerns. Machine learning as an approach to formulating and solving an IDS that is an effective solution for many IDS problems. However, related to machine learning, many challenging aspects need to be addressed when approaching an IDS. We can present them under the following three challenging and symmetric aspects in terms of effecting the performance: first, the concept drift [17]; second, high dimensionality [18]; and third, computational complexity [19]. Drift is caused by the continuous update of the statistical characteristics of the distribution of data generated from network traffic while conducting an attack, which is handled by updating the classifier. The property of high dimensionality is normal in Internet of Things (IoT) networks. The last one is the computational complexity caused by the considerable number of computations for performing learning updates of the classifier. This review is the first that presents IDS problems with the challenging aspects of machine learning as a solution for IDS.

The remaining of this article is organized as follows: In Section 2, we present the contribution; in Section 3, a review of the intrusion detection system is provided; next, in Section 4, we present the concept drift; in Section 5, high-dimensional aware methods are provided; in Section 6, computational efficient awareness models are provided; the datasets are given in Section 7; in Section 8, we present the challenges and discussion; and finally, the summary and conclusions are provided in Section 9.

2. Contributions

The aims of this study are to address the various stream data learning problems in general, as well as stream data-based attack detection and identification in the IoT. It provides the following contributions:

1. It is the first review article that tackles the problem of IDS from three perspectives, namely, concept drift, high dimensionality, and computational efficiency.
2. It discusses the evolving aspect of IDS attacks. It argues about the impact of the changes in the statistical distribution of the data and their corresponding classes, which requires developing concept drift-aware intrusion detection systems.
3. It focuses on reviewing the computational load of the approaches and their impact on the feasibility of applying them in real-world systems.
4. It provides a thorough discussion of the future challenges in IDS and the solutions that must be developed.

3. Intrusion Detection Systems

This section focuses on an IDS system. First, we present the definition of an IDS in Section 3.1, and then provide the taxonomy of an IDS in Section 3.2.

3.1. Definition

An intrusion detection system is comprised of an audit data collection agent that collects data on the system in question. Then, these data are either stored or processed directly by the detector and given to the site security office (SSO), followed by additional steps which usually start with further investigation of the reasons for the alarm.

The concept of an IDS appeared in earlier studies in the literature, starting from the work of [20] (computer network security monitoring and surveillance) who used some

tools to manage and review audit trails for detecting any abnormal activity in the network system. When considering an IDS, it is important to differentiate between two functions that exist in a typical IDS. The first function is intrusion alarming, which can detect any abnormal activity in the system. The second function is an SSO, which responds to the alarm and takes the appropriate action.

It is important to remember that an intrusion can take several different forms. An intruder could, for example, steal a password, and thus have the means to prove their identity to the machine. Such an intruder is known as a masquerader, and detecting such intruders is a significant problem in the field. People who are legal users of the system but misuse their rights and people who use pre-packaged exploit scripts, which are mostly available on the Internet, to target the system via a network, are often examples of intruders. Another example is the motion-based side-channel attack that aims to infer the character types on the smartphone interface using vibration-based predictions from three smartphone sensors, namely, gyro, accelerometer, and magnetometer [21]. Another type of attack is the Sybil attack when a node illegitimately identifies and threatens various types of networks such as wireless sensor networks [22]. This is by no means an exhaustive list, and the classification of threats to computer systems is a hot topic in academia and industry.

3.2. Intrusion Detection System Taxonomy

IDSs are categorized into three groups, i.e., anomaly-based detection, signature-based detection, and specification-based detection [23]. To summarize the taxonomy, we show a conceptual diagram in Figure 1.

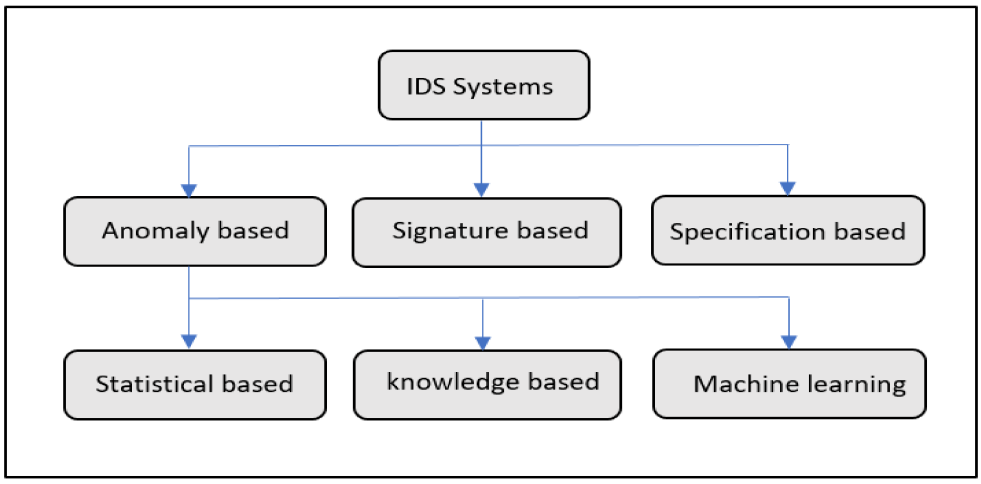

Figure 1. Taxonomy of intrusion detection system (IDS) systems.

1. Anomaly-Based Technique

Anomaly detection refers to the deviation of network traffic from its normal profile. The normal profile is captured in the network's non-attack conditions and is represented mostly by statistical data [24]. An example of such deviation is when a manager, who normally accesses the network in the daytime, uses his account to access it at night, which is regarded as a deviation of activity. Such a deviation is suspicious, and it can indicate an attack; however, such activity might not be associated with an attack. Thus, it is possible to have a false alarm based on this. Hence, continuous updates of the network's user activity patterns to avoid false alarms can be performed [25].

In this study, we are interested in an anomaly-based IDS. Therefore, we create the following sub-taxonomy of such systems:

Statistical-based anomaly IDS The statistical-based anomaly IDS matches the periodically captured statistical features from the traffic with a generated stochastic model of the normal operation or traffic [26]. The attack is reported as the deviation

between the two statistical patterns, i.e., the normal memorized one and the current captured one.

Knowledge-based anomaly IDS In knowledge-based anomaly detection, numerous rules are provided by experts in the form of an expert system or fuzzy-based system to define the behavior of normal connections and attacks. In fuzzy-based anomaly detection, the rule-based is connected to inputs. A subset of the rules is enabled based on the input values [27], sometimes heuristics or an UML-based description of the attack's behavior is provided [25].

Machine learning-based anomaly IDS An explicit or implicit model of the analyzed patterns is developed in a machine learning-based anomaly IDS. These models are revised regularly to boost intrusion detection efficiency based on past results. Sections 4 and 5 include more information and an analysis of the machine learning models for IDS.

2. Signature-Based Technique

A signature-based technique is also referred to as knowledge-based or misuse detection. It uses the signature of the attack and performs matching between the current traffic and the signature, and then reports an attack on the existence of matching, otherwise, it does not report an attack. Such an approach does not suffer from a high rate of false alarm like other approaches. However, it requires a continuous update of the signature [25].

3. Specification-Based Technique

A specification-based technique uses the specification or constraints to describe a certain program's operation and report any violation of such specification or constraints based on matching with the prior determined and memorized specification and constraints [28].

4. Concept Drift

The prediction should tolerate concept drift in the field that does not have prior knowledge for predicting concepts such as weather or finance. This happens when a noise at a certain time becomes information at another time or vice versa, as proposed by [29]. For a more formatted definition, we use the work of [30].

Assuming we have a stream data $D = (x, y)$, where x denotes the feature and y denotes the class label. The definition of concept drift is the change of the joint probability $p_t(x, y) = p_{t+1}(x, y)$. According to the Bayes theorem $p(x, y) = p(x)p(y|x)$, the change might occur because of drifting in $p(x)$ over time, or it might occur when the prior probability changes with time $p_t(y)$. The third reason for concept drift is a change in the posterior probability $p_t(y|x)$.

Concept Drift Aware Machine Learning Models

One of the typical problems of machine learning models is when the model falls into concept drift, which is due to the dynamic characteristics of real-world data and the non-stationary nature of its processes [31]. In the context of intrusion detection, the concept refers to the process of the predicted value of the state of the traffic connection or record [4]. It can be predicted as normal, or an attack, based on input variables that represent the features. The issue of concept drift happens due to changing the behavior of conducting the attack and inventing new methods or ways. This makes the problem of attack identification highly dynamic and characterized as a non-stationary process. Some researchers have described concept drift in an IDS as sudden concept drift [32]. In addition, researchers have considered IoT data to be subject to concept drift [33]. Some researchers have developed solutions for special cases of concept drift, for example, when the number of clusters changes [34].

The literature contains numerous approaches for dealing with the problem of concept drift. Some researchers [35] have proposed preparing a pool of classifiers, each trained on certain concepts and adopting a dynamic or time-based selection to subset them according to the test sample to counter the concept drift. Other researchers have proposed dealing

with concept drift in the same clustering model instead of adopting a separate block to deal with it. For example, [36] proposed an algorithm that maintains and updates online micro-cluster to distinguish evolution and concept drift from noisy data. Adopting online approaches to deal with concept drift has motivated researchers to propose an alternative model with an online nature to replace the offline models used in clustering and classification.

An example is by [37], where the concept drift has been reflected online to obtain the principal component analysis's eigenvalues. Overall, the general trend of studies in the literature is to develop clustering and classification models for network security and IDS, in particular, for handling the evolving nature of random variables and processes through dealing with concept drift. Unfortunately, IDS literature for handling concept drift is still non-significant and needs lots of developments. In the work of [38], an approach for detecting concept drift caused by dynamical significant feature changes was proposed. The concept drift detection is based on real-time feature selection using tracking adaptive statistical summaries of the data and class label distributions. In the work of [39], an online adaptation of an ensemble of deep neural networks was used to counter the concept drift and enable steady long-term performance. However, this work faces an issue of computational complexity. Table 1 shows the previous approaches for handling concept drift.

Table 1. Existing approaches for handling concept drift.

Reference	Description	Limitation
[35]	According to the test sample, a pool of classifiers is trained on certain concepts and adopts a dynamic or time-based selection to subset them according to the test sample to counter the concept drift.	It assumes that prior knowledge of the concept is not a valid assumption in the practical world.
[36]	An algorithm that maintains and updates online micro-cluster to distinguish evolution and concept drift from noisy data.	It handles only clustering.
[37]	Concept drift has been reflected in an online way to obtain the principal component's analysis's eigenvalues.	It assumes concept can be captured by data reduction only, which is not always true.
[38]	Concept drift detection based on real-time feature selection using tracking adaptive statistical summaries of the data and class label distributions	It is limited to only one type of concept drift, i.e., feature changing caused concept drift.
[39]	An online deep neural network model relies on an ensemble of varying depth neural networks that cooperate and compete to enable the model to steadily learn and adapt as new data, allowing for stable and long-lasting learning.	Concern about the computational complexity.

5. High Dimensional Aware Machine Learning

High dimensionality is one issue in machine learning in general, and in IDS applications in particular. High dimensionality becomes more challenging when the data are streamed due to the inability of storing the data to perform an analysis [40]. Some researchers have described the high dimensionality of IDS data as the curse of dimensionality [41] and have adopted various approaches to solve the issue. The concepts of core mini-clusters and grid are useful for summarizing the data, which provides a smaller version of the data, making it feasible to provide storage and computation for clustering [42]. Other researchers have also used separated approaches for reducing data, for example, [43,44]. An existing challenge is how to handle high dimensionality without affecting the quality of clustering. The concept of grid projection as a stage of data reduction was initially proposed by [45]. It separates the data space along all dimensions into intervals, and instead of storing the data, a statistical summary of the data projection in each interval is stored.

One of the most popular approaches, which used grid mapping, is the work of D-Stream [46], where a framework for clustering stream data using a density-based approach was developed. The framework proposed two phases, i.e., online and offline. The online phase's goal is to map the data into a grid, while the offline phase's goal is to compute the final clusters based on the grids. Micro-clusters as another stage of data reduction have

also been proposed in this framework, which was initially proposed by [47]. It builds small data structures that summarize the density and distribution of data points in space, called micro-clusters. Other researchers have extended the micro-cluster concept from a single level to multiple levels or hierarchical [48]. To support data stream clustering, a hierarchical tree is presented to dynamically store the micro-clusters at different granularity levels, depending upon the data arrival time. Various recent algorithms have relied on the concept of micro-clusters and integrated them with optimization algorithms. In [49], the authors proposed an ant colony stream clustering (ACSC) that is based on identifying a group of micro-clusters. The algorithm uses a tumbling window model and stochastic method to find rough clusters. Next, the rough clusters are refined using ant colony optimization. Astudy by [50] also used the concept of micro-clusters to update the micro-cluster radius recursively with an approach that uses a buffer for storing and filtering out irrelevant micro-clusters. Furthermore, their algorithm used an energy updating function based on the data stream's spatial information.

Some of the frameworks have focused on stream speed and handling different speeds of the data stream. For example, in the work of [51], a framework for stream data clustering, named as ClusTree, was proposed to handle different speeds of the stream. It also uses the concept of micro-clusters based on statistical modeling (mean and variance). The framework has been extended by [52] to the distributed framework. Another way of dealing with high dimensional data, grids, and the micro-cluster concept is the principal component analysis, used by numerous researchers. Table 2 shows the previous approaches for handling the issue of high dimensionality.

Table 2. Existing approaches for handling high dimensionality.

Reference	Description	Limitation
[42]	The concepts of core mini-clusters and grid are useful to summarize the data, which provides a smaller version for the data to make it feasible in providing the storage and computation for clustering.	It is still limited in the case of high dimensional data.
[48]	Extended the concept of micro-clusters from single level to multiple levels or hierarchical.	It causes a complicated architecture of storing the data.
[49]	An ant colony stream clustering (ACSC) is based on identifying a group of micro-clusters. The algorithm uses a tumbling window model and stochastic method to find rough clusters. Next, the rough clusters are refined using ant colony optimization.	It has a computational concern because of running the optimization inside the clustering.
[50]	Used the micro-cluster concept and updated the micro-cluster radius recursively with an approach that uses a buffer for storing and filtering out irrelevant micro-clusters. Furthermore, their algorithm used an energy updating function based on the spatial information of the data stream.	It is useful for reducing computation more than memory consumption.
[51]	A framework for stream data clustering, named as ClusTree, was proposed to handle different speeds of the stream. It also uses the concept of micro-clusters based on statistical modeling (mean and variance).	It assumes a normal distribution of data, which is not a valid assumption in all real-world problems.

6. Computational Efficient Machine Learning

One of the issues of an intrusion detection system based on intelligent algorithms is the real-time constraint, which requires adopting computationally light approaches to enable fast prediction. The literature has tackled this aspect but without a big focus. The majority of the approaches were evaluated from the accuracy of prediction perspective without reporting the execution time of both the learning and the prediction. In addition, while many approaches aim to achieve real-time performance by enabling parallel execution, such as [53], they have not focused on modifying the inside algorithm to make it computationally lighter. Other researchers have improved the real-time aspect of the ensemble learning-based IDS. The real-time in such type an IDS is affected by feedback waiting time for another IDS when a consultation is made. Building a neural network trained on partial IDS feedback predicts the feedback and makes the decision making faster [54]. Other researchers have considered a lightweight IDS for enabling them on fog networks, such as [55], where a multi-layer perceptron model was used and execution on raspberry pi was

performed. Notably, the approach of building a lightweight IDS has mostly been used for IoT applications. For example, [56] used a support vector machine (SVM) assisted by two or three incomplete features for an IDS, yielding fast detection time. The authors have used this as an indicator of the algorithm's lightweight and the CPU execution time.

In [57], the authors proposed an artificial bee colony integrated with a lightweight IDS to detect Sybil attacks. The algorithm works in real-time to track the arrival time of control messages for low power and lossy network (RPL). However, such an algorithm is customized for the mentioned protocol, making it non-applicable for the general IDS system. In the work of [22], various methods for detecting Sybil attacks, such as radio resource testing, key validation for random key predistribution, position verification, and registration, were introduced.

To summarize, we present each of the approaches in Table 3, presenting their limitations.

Table 3. Existing approaches for handling computational complexity.

Reference	Scope	Contribution	Limitation
[54]	General IDS	Enabling parallel execution	Lacking focus on modifying the inside algorithm to make it computationally lighter
[55]	IDS	Ensemble learning based	Partial IDS feedback is not adequate in ensemble learning
[56]	Fog IDS	Multi-layer perceptron model was used and execution on raspberry pi was performed	Back-propagation training is iterative and requires time
[57]	IoT IDS	Support vector machine SVM assisted by two or three incomplete features	SVM is the most efficient classifier
[58]	Low power and lossy network (RPL) IDS	Tracking of the arrival time of control messages	Protocol dependent

7. Dataset

In this section, we look at some of the most well-known datasets in IDSs and IDSs for IoT. We provide each one with statistical details on the number of records, classes, and decomposition of those records.

7.1. KDD99 Dataset

The KDD99 dataset has been the most widely used dataset for network intrusion detection, since 1999; [57] generated this data collection based on the data collected during the DARPA IDS assessment program in 1998. DARPA'98 consists of approximately 4 gigabytes of compressed raw (binary) TCP dump data from seven weeks of network traffic, which can be processed into approximately five million link records, each containing about one hundred bytes. There are approximately two million link records in the two weeks of test results. Each of the nearly 4,900,000 single link vectors, in the KDD training dataset, has forty-one features and is classified as either regular or an attack of a specific kind.

1. A denial of service (DoS) attack occurs when an attacker makes the computing or memory resource too busy or complete to handle legitimate requests or denies a legitimate user access to a computer.
2. A user-to-root attack (U2R) is a type of exploit in which the attacker gains access to a system's regular user account (possibly through password sniffing, a dictionary attack, or social engineering) and exploits a vulnerability to gain root access.
3. A remote-to-local attack (R2L) occurs when an attacker can send packets to a computer over a network. Still, no account on that machine exploits a vulnerability to gain local access as that machine's user.
4. A probing attack is an effort to collect information about a network of computers to obfuscate its security controls.

Figure 2 shows the distribution of the classes based on the sample sizes in a bar graph. There is an imbalance in the dataset, as can be observed. The problem of classification or clustering becomes extremely difficult as a result of this imbalance.

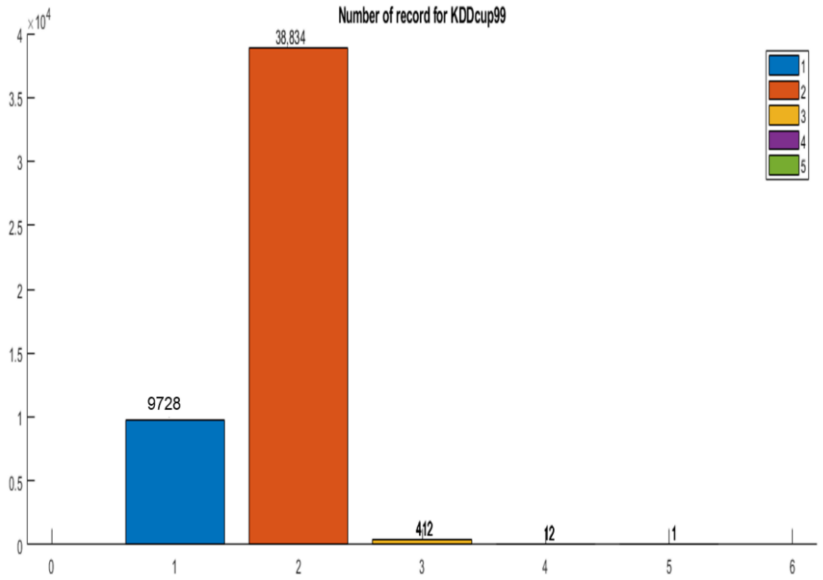

Figure 2. Classes' distribution in terms of sample sizes for the KDD99 dataset.

7.2. NSL Database

The NSL-KDD dataset [58], consisting of selected records from the entire KDD dataset, is proposed. The benefit of the NSL-KDD dataset is the statistical analysis revealed significant problems in the dataset that have a significant impact on system efficiency. This includes redundant records in the train set, duplicate records in the test set, and the number of selected records from each difficult level category not related to the percentage of records in the original KDD dataset. In NSL, certain problems were resolved.

Out of the 37 attacks present in the test dataset, 21 are included in the training dataset. The recognized attack forms are those found in the training dataset, while the novel attacks are not found in the training dataset.

We provide the classes' distribution in the data based on the sample sizes in the bar graph provided in Figure 3. It can be observed that the classes are distributed non-equally in terms of the number of samples.

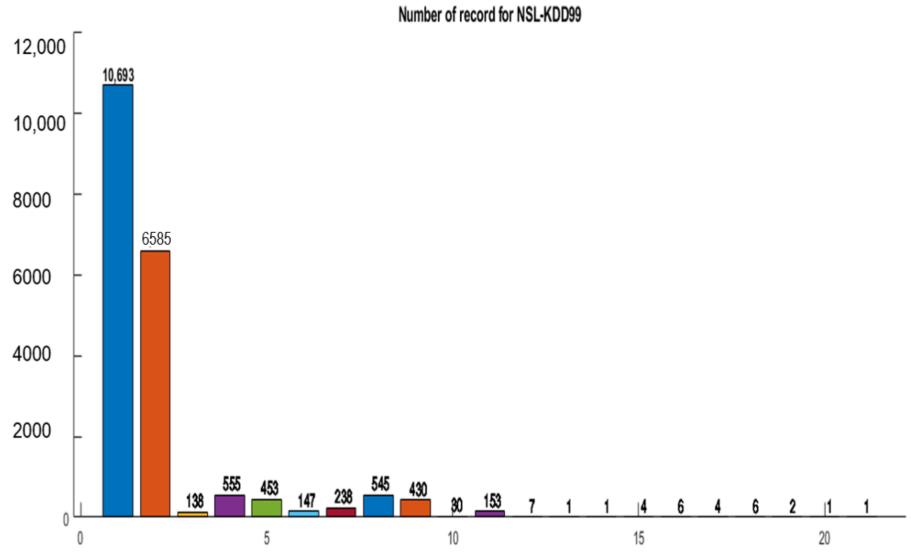

Figure 3. Classes' distribution in terms of sample sizes for the NSL dataset.

7.3. Kyoto Dataset

There are 24 statistical features in this dataset, i.e., 14 traditional features and ten additional features. On the basis of the KDD Cup 99 dataset [59], the first 14 features were extracted from the 41 original features using honeypot systems installed at Kyoto University. In addition to those 14 features, they have extracted an additional ten features, making them investigate what happens on the network more effectively.

The classes' distribution according to the sample sizes is provided in the bar graph in Figure 4. As we observe, the data is a combination of three classes; the largest size is 82%, the second largest being 16% of the total data, while the smallest one is 2%.

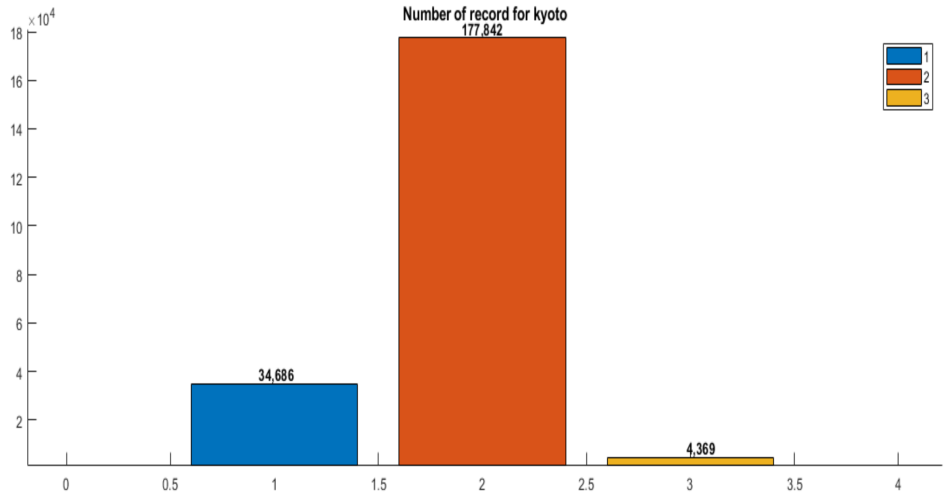

Figure 4. Classes' distribution in terms of samples sizes for the Kyoto dataset.

8. Challenges and Discussion

Reading the literature, we conclude that an IDS for the IoT based on stream data analysis requires dealing with three main challenges:

(1) In concept drift, attacks are not conducted using the same way. Hence, it is needed to handle their evolving aspect. The evolving aspects of attacks imply changes in the statistical distribution of the data and their corresponding classes. Such change is named concept drift. The approaches for solving concept drift can be summarized as follows: (1) They assume that prior knowledge of the concept is not a valid assumption in a practical world. (2) They also assumes concepts can be captured by data reduction only, which is not always true. (3) Some of them do not handle sequential classification, which is an essential part of IDS theory.

(2) High dimensionality IoT-based systems are categorized as high-dimensional systems, and therefore the issue of high dimensionality must be handled in IDSs for the IoT. The approaches developed in the literature for high dimensionality suffer from the following: (1) They can cause a complicated architecture of storing the data. (2) Moreover, they have only considered the computational aspect of analyzing high-dimensional data with less attention to memory consumption. (3) They also assume a normal distribution of data, which is not a valid assumption in all real-world problems.

(3) One significant issue of IDSs for the IoT systems is computational concerns. Studies in the literature have taken numerous approaches for addressing this concern. However, they still suffer from a lack of focus on modifying the inside algorithm to make it computationally lighter with less attention for the iterative training approaches such as backpropagation.

9. Summary and Conclusions

In this article, we presented a literature survey on the topic of an intrusion detection system (IDS) and its challenges. The focus of the article was on using machine learning

for a IDS in the Internet of Things. Hence, we investigated three main challenges of machine learning when dealing with an IDS for the IoT, i.e., evolving and concept drift, high dimensionality, and computational complexity. We dedicated a separate section for presenting each of these challenges in general, and their relationships with machine learning in particular. Furthermore, we presented three main datasets, namely KDD99, NSL, and Kyoto. Finally, we presented the main challenges in the IoT and IDSs and approaches for dealing with them according to the existing literature on this topic.

Author Contributions: Conceptualization, A.A. (Ahmed Adnan), A.M., A.A.A.G., A.A. (Azizol Abdullah), and F.H.; writing—original draft preparation, A.A. (Ahmed Adnan); supervision, A.M., A.A.A.G., A.A. (Azizol Abdullah), and F.H.; funding acquisition, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ogudo, K.A.; Nestor, D.M.J.; Khalaf, O.I.; Kasmaei, H.D. A device performance and data analytics concept for smartphones' IoT services and machine-type communication in cellular networks. *Symmetry* **2019**, *11*, 593. [CrossRef]
2. Darwish, A.; Hassanien, A.E.; Elhoseny, M.; Sangaiah, A.K.; Muhammad, K. The impact of the hybrid platform of internet of things and cloud computing on healthcare systems: Opportunities, challenges, and open problems. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 4151–4166. [CrossRef]
3. Rehman, M.H.u.; Yaqoob, I.; Salah, K.; Imran, M.; Jayaraman, P.P.; Perera, C. The role of big data analytics in industrial Internet of Things. *Future Gener. Comput. Syst.* **2019**, *99*, 247–259. [CrossRef]
4. Zolanvari, M.; Teixeira, M.A.; Gupta, L.; Khan, K.M.; Jain, R. Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 6822–6834. [CrossRef]
5. Teng, H.; Liu, Y.; Liu, A.; Xiong, N.N.; Cai, Z.; Wang, T. A novel code data dissemination scheme for Internet of Things through mobile vehicle of smart cities. *Future Gener. Comput. Syst.* **2019**, *94*, 351–367. [CrossRef]
6. Muthuramalingam, S.; Bharathi, A.; Kumar, S.R.; Gayathri, N.; Sathiyaraj, R.; Balamurugan, B. Iot based intelligent transportation system (IoT-its) for global perspective: A case study. *Intell. Syst. Ref. Libr.* **2019**, *154*, 279–300.
7. Kraemer, F.A.; Ammar, D.; Braten, A.E.; Tamkittikhun, N.; Palma, D. Solar energy prediction for constrained IoT nodes based on public weather forecasts. In Proceedings of the Seventh International Conference on the Internet of Things, Linz, Austria, 22–25 October 2017.
8. Helbing, D. Societal, Economic, Ethical and Legal Challenges of the Digital Revolution: From Big Data to Deep Learning, Artificial Intelligence, and Manipulative Technologies. In *Towards Digital Enlightenment*; Springer: Cham, Switzerland, 2015.
9. Nabi, R.M.; Saeed, S.A.M.; Haron, H. Artificial intelligence techniques and external factors used in crime forecasting in violence and property: A review. *J. Comput. Sci.* **2020**, *16*, 167–182. [CrossRef]
10. Al-Naeem, M.; Rahman, M.A.; Ibrahim, A.A.B.; Rahman, M.M.H. AI-based techniques for DDoS attack detection in WSN: A systematic literature review. *J. Comput. Sci.* **2020**, *16*, 848–855. [CrossRef]
11. Mahdavinejad, M.S.; Rezvan, M.; Barekatain, M.; Adibi, P.; Barnaghi, P.; Sheth, A.P. Machine Learning for Internet of Things Data Analysis. *Digit. Commun. Netw.* **2017**, *4*, 161–175. [CrossRef]
12. Xiao, L.; Wan, X.; Lu, X.; Zhang, Y.; Wu, D. IoT Security Techniques Based on Machine Learning. *IEEE Signal Process. Mag.* **2018**, *35*, 41–49. [CrossRef]
13. Diro, A.A.; Chilamkurti, N. Distributed Attack Detection Scheme using Deep Learning Approach for Internet of Things. *Future Gener. Comput. Syst.* **2017**, *82*, 761–768. [CrossRef]
14. Zhao, Q.; Chen, S.; Liu, Z.; Baker, T.; Zhang, Y. Blockchain-based privacy-preserving remote data integrity checking scheme for IoT information systems. *Inf. Process. Manag.* **2020**, *57*, 102355. [CrossRef]
15. Hu, Y.; Ren, P.; Luo, W.; Zhan, P.; Li, X. Multi-resolution representation with recurrent neural networks application for streaming time series in IoT. *Comput. Netw.* **2019**, *152*, 114–132. [CrossRef]
16. Leech, C.; Raykov, Y.P.; Ozer, E.; Merrett, G.V. Real-time room occupancy estimation with Bayesian machine learning using a single PIR sensor and microcontroller. In *2017 IEEE Sensors Applications Symposium (SAS)*; IEEE: Hoboken, NJ, USA, 2017.
17. Iwashita, A.S. An Overview on Concept Drift Learning. *IEEE Access* **2019**, *7*, 1532–1547. [CrossRef]
18. Ghaddar, B.; Naoum-Sawaya, J. High dimensional data classification and feature selection using support vector machines. *Eur. J. Oper. Res.* **2018**, *265*, 993–1004. [CrossRef]

19. Al-yaseen, W.L.; Ali, Z.; Zakree, M.; Nazri, A. Real-time multi-agent system for an adaptive intrusion detection system. *Pattern Recognit. Lett.* **2017**, *85*, 56–64. [CrossRef]
20. Anderson, J.P. Computer security threat monitoring and surveillance. In *Technical Report*; James P Anderson Company: Fort Washingt, PA, USA, 1980; p. 56.
21. Javed, A.R.; Beg, M.O.; Asim, M.; Baker, T.; Al-Bayatti, A.H. AlphaLogger: Detecting motion-based side-channel attack using smartphone keystrokes. *J. Ambient Intell. Humaniz. Comput.* **2020**, 0123456789. [CrossRef]
22. Newsome, J.; Shi, E.; Song, D.; Perrig, A. The Sybil attack in sensor networks: Analysis & defenses. In Proceedings of the Third International Symposium on Information Processing in Sensor Networks IPSN, Berkeley, CA, USA, 27 April 2004; pp. 259–268.
23. Liao, H.; Lin, C.R.; Lin, Y.; Tung, K. Journal of Network and Computer Applications Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* **2013**, *36*, 16–24. [CrossRef]
24. Fernandes, G.; Rodrigues, J.J.; Carvalho, L.F.; Al-Muhtadi, J.F.; Proença, M.L. A comprehensive survey on network anomaly detection. *Telecommun. Syst.* **2018**, *70*, 447–489. [CrossRef]
25. Hamamoto, A.H.; Carvalho, L.F.; Sampaio LD, H.; Abrão, T.; Proença, M.L., Jr. Network anomaly detection system using genetic algorithm and fuzzy logic. *Expert Syst. Appl.* **2018**, *92*, 390–402. [CrossRef]
26. Kabir, E.; Hu, J.; Wang, H.; Zhuo, G. A novel statistical technique for intrusion detection systems. *Future Gener. Comput. Syst.* **2017**, *79*, 303–318.
27. Petkovic, M.; Basicevic, I.; Kukolj, D.; Popovic, M. Evaluation of takagi-sugeno-kang fuzzy method in entropy-based detection of DDoS attacks. *Comput. Sci. Inf. Syst.* **2018**, *15*, 139–162. [CrossRef]
28. Dupont, G.; den Hartog, J.; Etalle, S.; Lekidis, A. Network intrusion detection systems for in-vehicle network—Technical report. *arXiv* **2019**, arXiv:1905.11587.
29. Schlimmer, J.C.; Granger, R.H. Incremental learning from noisy data. *Mach. Learn.* **1986**, *1*, 317–354. [CrossRef]
30. Priya, S.; Uthra, R.A. Comprehensive analysis for class imbalance data with concept drift using ensemble based classification. *J. Ambient Intell. Humaniz. Comput.* **2020**. [CrossRef]
31. Webb, G.I.; Hyde, R. Characterizing Concept Drift. *Data Min. Knowl. Discov.* **2016**, *30*, 964–994. [CrossRef]
32. Ahmadi, Z.; Kramer, S. Modeling recurring concepts in data streams: A graph-based framework. *Knowl. Inf. Syst.* **2017**, *55*, 15–44. [CrossRef]
33. Stolpe, M. The Internet of Things: Opportunities and Challenges for Distributed Data Analysis. *ACM SIGKDD Explor. Newsl.* **2016**, *18*, 15–34. [CrossRef]
34. De Andrade, J.; Raul, E.; Gama, J. An evolutionary algorithm for clustering data streams with a variable number of clusters. *Expert Syst. Appl.* **2017**, *67*, 228–238. [CrossRef]
35. Almeida, P.R.L.; Oliveira, L.S.; Britto, A.S.; Sabourin, R. Adapting dynamic classifier selection for concept drift. *Expert Syst. Appl.* **2018**, *104*, 67–85. [CrossRef]
36. Din, S.U.; Shao, J. Exploiting evolving micro-clusters for data stream classification with emerging class detection. *Inf. Sci.* **2020**, *507*, 404–420. [CrossRef]
37. Park, S.; Kim, J. Network Intrusion Detection through Online Transformation of Eigenvector Reflecting Concept Drift. In Proceedings of the International Conference on Data Science, E-Learning and Information Systems, Madrid, Spain, 1–2 October 2018; pp. 2–5.
38. Hammoodi, M.S.; Stahl, F.; Badii, A. Real-time feature selection technique with concept drift detection using adaptive micro-clusters for data stream mining. *Knowl. Based Syst.* **2018**, *161*, 205–239. [CrossRef]
39. Wahab, O.A. Sustaining the Effectiveness of IoT-Driven Intrusion Detection over Time: Defeating Concept and Data Drifts. pp. 1–10. Available online: https://www.techrxiv.org/articles/preprint/Sustaining_the_Effectiveness_of_IoT-Driven_Intrusion_Detection_over_Time_Defeating_Concept_and_Data_Drifts/13669199/1 (accessed on 26 February 2021).
40. Braverman, V. Clustering High Dimensional Dynamic Data Streams. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
41. Yin, C.; Xia, L.; Zhang, S.; Sun, R.; Wang, J. Improved clustering algorithm based on high-speed network data stream. *Soft Comput.* **2017**, *22*, 4185–4195. [CrossRef]
42. Amini, A.; Saboohi, H.; Herawan, T.; Wah, T.Y. MuDi-Stream: A multi density clustering algorithm for evolving data stream. *J. Netw. Comput. Appl.* **2016**, *59*, 370–385. [CrossRef]
43. Gao, X.; Shan, C.; Hu, C.; Niu, Z.; Liu, Z. An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access* **2019**, *7*, 82512–82521. [CrossRef]
44. Jaber, A.N.; Zolkipli, M.F.; Shakir, H.A.; Mohammed, R. Host Based Intrusion Detection and Prevention Model Against DDoS Attack in Cloud Computing. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*; Springer: Cham, Swizerland, 2018.
45. Gao, J.; Li, J.; Zhang, Z.; Tan, P.N. An incremental data stream clustering algorithm based on dense units detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 420–425.
46. Chen, Y.; Tu, L. Density-based clustering for real-time stream data. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, 13–17 August 2007; pp. 133–142.
47. Aggarwal, C.C.; Ctr, T.J.W.R.; Han, J.; Wang, J.; Yu, P.S.; Ctr, T.J.W.R. A Framework for Clustering Evolving Data Streams. In Proceedings of the 2003 VLDB Conference, Berlin, Germany, 9–12 September 2003.

48. Shao, J.; Tan, Y.; Gao, L.; Yang, Q.; Plant, C.; Assent, I. Synchronization-based clustering on evolving data stream. *Inf. Sci.* **2019**, *501*, 573–587. [CrossRef]
49. Fahy, C.; Yang, S.; Gongora, M. Ant Colony Stream Clustering: A Fast Density Clustering Algorithm for Dynamic Data Streams. *IEEE Trans. Cybern.* **2019**, *49*, 2215–2228. [CrossRef] [PubMed]
50. Islam, M.K.; Ahmed, M.M.; Zamli, K.Z. A buffer-based online clustering for evolving data stream. *Inf. Sci.* **2019**, *489*, 113–135. [CrossRef]
51. Kranen, P.; Assent, I.; Baldauf, C.; Seidl, T. The ClusTree: Indexing micro-clusters for anytime stream mining. *Knowl. Inf. Syst.* **2011**, *29*, 249–272. [CrossRef]
52. Hesabi, Z.R.; Sellis, T.; Liao, K. *DistClusTree: A Framework for Distributed Stream Clustering*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 1.
53. Sharma, J.; Giri, C.; Granmo, O.C.; Goodwin, M. Multi-layer intrusion detection system with ExtraTrees feature selection, extreme learning machine ensemble, and softmax aggregation. *EURASIP J. Inf. Secur.* **2019**, *2019*, 1–16. [CrossRef]
54. Abusitta, A.; Bellaiche, M.; Dagenais, M.; Halabi, T. A deep learning approach for proactive multi-cloud cooperative intrusion detection system. *Future Gener. Comput. Syst.* **2019**, *98*, 308–318. [CrossRef]
55. Khater, B.S.; Wahab, A.W.B.A.; Idris, M.Y.I.B.; Hussain, M.A.; Ibrahim, A.A. A lightweight perceptron-based intrusion detection system for fog computing. *Appl. Sci.* **2019**, *9*, 178. [CrossRef]
56. Jan, S.U.; Ahmed, S.; Shakhov, V.; Koo, I. Toward a Lightweight Intrusion Detection System for the Internet of Things. *IEEE Access* **2019**, *7*, 42450–42471. [CrossRef]
57. Murali, S.; Jamalipour, A. A Lightweight Intrusion Detection for Sybil Attack under Mobile RPL in the Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 379–388. [CrossRef]
58. Rummel, M.; Rummel, M. *"Der Social Entrepreneurship-Diskurs. Eine Einführung in die Thematik," Wer Sind Soc. Entrep. Deutschland?* Springer: Berlin/Heidelberg, Germany, 2011; pp. 21–38. [CrossRef]
59. Song, J.; Takakura, H.; Okabe, Y.; Eto, M.; Inoue, D.; Nakao, K. Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation. In Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, Salzburg, Austria, 10 April 2011; pp. 29–36.