International Journal of Intelligent Computing and Technology

IJICT

ISSN: 2457-0249 (Online)

Website: https://www.ijict.com Jan (2022) - Vol.5, Iss.2 Pages:39-54

FUTURE TRENDS IN NETWORK BASED INTRUSION DETECTION SYSTEMS

T S Ramya[1], N Subha Raagavi[2], Dr N Karpagavallii[3]

[1,2]BCA, [3]Assistant Professor, PG Department of Computer Science

Holy Cross College, Trichy, Tamilnadu

Article History- Received: June 2021; Published: Jan 2022

Abstract

In the security of network, an Intrusion Detection System (IDS) plays a major role, which is used to predict whether the traffic is normal or abnormal. The major challenge is to develop an effective Network based IDS as NIDS for identifying the attack situations. The best solution for modelling the efficient NIDS is to implement it with machine learning (ML) techniques using advanced intrusion datasets. This article gives you a brief overview of tagged intrusion datasets and mostly commonly used ML techniques. Next, it provides a brief overview of the literature on ML techniques applied to implement NIDS using different datasets for finding whether the traffic is normal or attack. In order to identify the current challenges and future trends, the integration of various datasets with ML techniques are presented in this article. The problems associated with NIDS is also explained in this review study. It will reveals the future development of effective NIDS model by improving the existing NIDS techniques.

Keywords: *Intrusion detection, anomaly, intrusion dataset, machine learning and security*

INTRODUCTION

Nowadays, it is difficult to imagine a world without the Internet. Everyone depends on the Internet. It has become a major model in various applications such as education and business. This is why the security of data transmitted over the Internet is essential. Therefore the secure network is maintained by the IDS. IDS closely monitors traffic and identifies it as regular or spam. Today, most applications are based on advanced networking technologies, such as wireless networks, wireless sensor networks, and Bluetooth. In the case of wireless sensor networks, security systems such as key management protocols, authentication techniques, and security protocols cannot be used due to resource limitations. The IDSis an ideal security feature for wireless sensor networks.

IDS is a security system used to monitor abnormal network behavior [1-2]. IDS identifies and notifies whether user activity is normal or not. An ID compares user activity with already stored intrusion logs to detect intrusion. Accurate prediction models for large data sets can be generated using supervised ML techniques which cannot be used with traditional methods.

As defined by Tom Mitchell [3], ML-based intrusion detection is divided into two categories: malpractice and abuse. IDS patterns are learned from training data, so an abuse-based approach is used. Absorption-based detection can only detect known attacks, and new attacks cannot be detected. Anomaly-based IDS monitors normal behavior and considers any change in behavior as an anomaly. Therefore, IDS-based network is able to detect new attacks that have not been learned from the training model. So far, various ML techniques have been proposed, such as artificial neural networks [4], Support Vector Machine (SVM), and Naïve Bayes [5-6], which are techniques based on intrusion detection. The author from [7] proposed a new invention that combines different techniques, which is called as a hybrid detection technique. The literature comparing supervised MLtechniques in intrusion detection is limited. Therefore, this article aims to understand the implications of using MLtechniques in intrusion detection.

Computer networks are vulnerable to attacks if do not have a security plan in an appropriate place. Hence the importance of this paper is multifold as this paper focuses on the following points.

This paper discusses some popular and latest ML algorithms to reveal their characteristics and limitations. This will help the researchers to select an appropriate algorithm for carrying out their research. It describes commonly used intrusion datasets. Periodic assessment of intrusion datasets plays a vital role in attaining NIDS goals. It helps in selecting the appropriate dataset for the evaluation of a specific NIDS. This also helps in dataset enhancement.

A periodic assessment of existing intrusion detection models is necessary to reveal the recent advancement and challenges in NIDS modeling. A deep analysis of different network domains is performed to unfold their security concerns and limitations that will aid in enhancing the performance of existing NIDS and implementing new improved models. FP, FN, data imbalance, etc., are common problems that degrade NIDS performance. These problems are discussed to attract the attention of the researchers so that they can gain important insights into how to improve NIDS performance.

The review has the following sections: the importance of intrustion dataset is given in Section 2, the review of ML techniques is presented in Section 3. Security on ML is described in Section 4. The related works of ML that are used for identifying the attacks in NIDS is provided in Section 5. The problems associated with NIDS is given in Section 6, finally the conclusion of the study is depicted in Section 7.

PUBLICLY AVAILABLE INTRUSION DATASETS

Intrusion datasets are surveyed and analyzed in several existing literature works with different objectives [8-10]. This section describes publicly available intrusion datasets namely KDD Cup '99, Network Security Laboratory-KDD (NSL-KDD), Aegean Wi-Fi Intrusion Dataset (AWID), Yahoo Webscope S5 anomaly benchmark, Numenta Anomaly Benchmark (NAB), Kyoto 2006+, UNSW-NB 15, BoT_IoT, Drebin, Contagio, and Genome. Here, we will discuss some of the major publically available datasets.

KDD Cup '99

In 1998, the Defense Advanced Research Projects Agency (DARPA), a division of the U.S. Defense Unit conducted an evaluation program in MIT Lincoln Labs to serve the objective of examining intrusion detection researches [11]. An extensive range of intrusion attack traffic was simulated in the U.S. Air Force LAN environment. KDD cup 99 comprises a set of these traffics [12]. KDD Cup '99 dataset has a total of 41 attributes and one more field namely attack class that labels all the observations into normal or the attacks that fall under one of the four categories: Denial-of-Service (DoS), Remote to Local (R2L), User to Root (U2R), and Probing/surveillance. KDD Cup '99 dataset resulted in biased classification due to some inherent flaws in the dataset such as redundancy and missing values in observations [13].

NSL-KDD dataset

It is an improvement over the KDD Cup '99 dataset in which

- ❖ Insignificant observations are removed from the training dataset. This resulted in unbiased classifier generation towards the more frequent record.
- ❖ Duplicate records are removed in the test sets. This resulted in unbiased learners' performance towards the approaches that otherwise better classify frequent records only.
- ❖ KDD dataset records have a different degree of difficulty. For the preparation of NSL-KDD, records are selected in inverse proportion to the percentage of records in the whole dataset. This resulted in efficient evaluation accuracy of diverse learning methods.
- ❖ A fair quantity of train and test dataset records in NSL-KDD leads to the efficient execution of experiments, Consistent and comparable evaluation results.

The attack labels count and the attributes count in the NSL-KDD dataset are similar to those in the KDD Cup '99 dataset [14]. But this new version also suffers from the problems discussed in [15]. NSLKDD doesn't provide exact definitions of the attacks and doesn't represent existing real networks. Its compatibility with real network traffic is not verified. Despite the flaws, KDD Cup '99 and NSLKDD datasets are still being used in many recent intrusion detection research works [16-17].

AWID dataset

AWID Offers tools, methodologies to implement wireless network IDS. AWID dataset comprises WLAN traffic in packet-based format. AWID comprises two versions: Large dataset and Reduced dataset, which are further subdivided into a high-level labeled dataset and Finer grained labeled dataset. This dataset has 155 features, including the class label [18]. AWID is imbalanced. So it needs proper pre-processing before use.

UNSW NB-15 dataset

UNSW NB-15 dataset comprises real recent normal network traffic traces as well as recent synthesized anomalous traffic activities [19-20]. This dataset includes a total of 2,540,038 flows out of which 2,218,755 are legitimate flows and 321,283 are attack flows. It has a total of 49 features including class labels. Many additional features are suitable for the detection of new types of attacks. Contemporary low footprint attacks are eventually reflected by the attack groups.

BoT_IoT dataset

BoT_IoT dataset includes real and simulated IoT network traffic [21]. The traffic comprises ordinary traffic and botnet traffic with 73,370,443 records. The BoT_IoT dataset has a total of 46 features including three class labels namely 'attack,' 'category,' and 'subcategory.' The 'attack' class label has two values; '0' for normal traffic and '1' for attack traffic. The 'category' class label divides the attack traffics into 3 categories which are further subdivided into 6 subcategories by the 'subcategory' class label.

Malware datasets- Drebin, Contagio, and Genome

Drebin, Contagio, and Genome are popular malware datasets, which are used to detect and classify widespread malware [22]. Genome dataset contains different types of Android malware (Collection duration: August 2010- October 2011) [23]. Drebin dataset contains real Android malware and real Android application samples from different websites (Collection duration: August 2010- October 2012) [24]. The Contagio dataset contains mobile malware samples as well as benign samples. This dataset is online accessible at Contagio Malware Dump [25].

The behaviour and limitations of the ML method must be known before applying it for NIDS modelling using any particular dataset. The next section discusses some ML methods popular in NIDS modelling.

MACHINE LEARNING TECHNIQUES

The combination of computer science and artificial intelligence presents the ML, which uses the particular data and programming languages to study the algorithms. In other words, ML is a process that are used to understand, study and predict the human beings's world by computer systems or machines. "MLis the study of how machines acquire new knowledge and skills and reorganize existing knowledge” [26-29]. At the time of implementation of ML, people conducted research to allow machines to learn, acquire skills, and build their own world of knowledge automatically. Subsequently, the term "ML" was explicitly coined by Samuel in 1959, [30] that are formed from the artificial intelligence study includes computational learning and pattern recognition theory. The computers are allowed to gain experience and modify respectively, which is the main concept of ML.

Data plays an important role in ML. Data patterns define learning outcomes. MLfirst requires data entry, also known as samples, training sets, and cases. With the help of the provided data sets, the machine reconstructs its internal relationships, which are the result of "learning" (also known as "training"), and presents the acquired knowledge through certain forms of output, such as identification, classification, and prediction. (also known as a "manual"). Specifically, regression models generate a mathematical variable; Taxonomic models create a taxonomic variable, etc.

According to the learning characteristics of provided datasets, the ML is categorized into three learning models such as supervised, semi-supervised and unsupervised learnings [31].

In the goal of ML algorithms, a model is created for mapping the inputs and outputs, when the attributes of input and output datasets are completely classified and this is called supervised learning. The classification and regression are included in the representative applications, here two mostly common used supervised ML algorithms are discussed:

SVM: In order to perform binary classification, SVM is used. There are two categories presented in this algorithm, where each data labels belongs to any one of these two categories with the series of training data. In the SVM training algorithm, a non-probabilistic binary linear classifier is constructed by arranging a new data into that categories. A set of hyperplanes are used in the feature space between two classes in the SVM model. In ordeer to classify the inconsistent sensor data with high dimensional features, SVM is the most suitable technique.

Neural Network (NN): A set of three layers such as input, hidden and output layers are presented in the large and complex network called NN. A vast amount of neurons are provided in each layer. In the previous layer, neurons' output is obtained by the inputs of the neurons to the current layer. A feedback or observations are used to learn the entire network's specific parameters via training datasets using NN. However, the network structure has a high execution time and low local problems, because of its complex structure.

In the unsupervised study, there were no labels for data sets. Algorithms often distinguish their own characteristics and patterns. Generally, models make deep correlations with the help of internal inference, depending on the similarity or distance between data inputs. The example of unsupervised learning algorithms is called as clustering models, which does not contain any conducts or advices while compared to supervised learning that deals with pre-defined labels. In clustering, the classification of objects with same attributes will be put in the same group called cluster. In various applications, some of the typical clustering algorithms such as hierarchical clsutering and connectivity models are used, which is developed by using distance connectivity. In order to describe a class, single vector is used by centroid models called k-means algorithm. The data is manipulated with statistical distributions by distribution models called expectation-maximization algorithms. In many data fusion models, k-means is the most commonly used algorithm that is described as follows:

K methods are the most widely used aggregation methods which reveal a structure in data by reducing a specific target function. When n data is placed in a space of d dimensions, k points are initially randomly selected as centers of mass by calculating the distances between data center with its nearest center. The rearrangement of distribution plots and recalculation of group centers are used to achieve the local square error distortion and small distance, which is the main aim of optimization. K- means belongs to the group on the basis of heterogeneity. In fact, grouping is a difficult problem in NP, so there is no general solution. As shown in [32] and [33], there are some representative effective models for solving the problem that k refers to.

ML is considered as semi-supervised learning only, when the specific training dataset has incomplete labels. In this case, the sorted data entries play an important role in determining the range. A large set of unlabeled data inputs can help improve the accuracy of the decision bounds and the stability of the entire model. Some of the most commonly used ML techniques are described below.

Decision Tree (DT)

The DT algorithm is commonly used for classification problems. In this procedure, the dataset is examined and modelled. Therefore, when a fresh data thing is assigned for classification, the data obtained from the previous data set will be categorized accordingly. The DT algorithm can also be used to test for the emotions. For this cause, the procedure will also study and model the data. As an outcome, the model can organize which type of emotions depends on which model in which future data is generated. The power of DT can work with this huge dataset. It works well for real-time detection because DT provides the highest detection efficiency and can be replicated and simply explained. Another beneficial stuff of DT is its simplification accuracy [34].

Bayesian Networks

In this model that codes the probabilistic associations between variables of interest. This method is commonly used for hate speech detection in mixture with statistical projects. Training sets with the target class are supplied as part of the Nave Bayesian (NB) algorithm. Attribute values connected with class C are used to name the training set and characterise each attribute's value. A Bayesian method to classifying an invisible example is to allocate the most likely target class. $Cmap$Given the attribute values $(a1, a2 \ldots an)$ that define the example.$Cmap = argmaxCj\Sigma CPCja1, a2 \; \ldots \; \ldots \; \ldots)$The expression can be revision using this theorem as

$$Cmap = argmaxCj\Sigma \, (a1, a2 \; \ldots \; \ldots \; \ldots \; an \,|)(Cj) \qquad (1)$$

Each of $P()$ is easy to estimate by simply counting the number of times each target class Cj appears in the training set. Based on the simplified premise that the likelihood of noticing a1,a2...an is merely the sum of the possibilities for the separable qualities, the NB method is used: $P(a1, a2 \ldots \ldots \ldots an\,|Cj\,) \;= \bullet\; iP(ai|Cj\,)$.

Exchanging this into equation 2, we get

$$CNB \;=\; argmaxCj\Sigma\,(Cj\;\bullet\;i\,(ai)|Cj\,) \qquad (2)$$

The target class prequel is represented by an NB classifier. In a simple Bayesian procedure the probability standards of Eq. 2 are assessed from the training data. These approximate values are then used to categorize unidentified instances. A technique that offers numerous rewards, with the ability to rely on variable code and the ability to predict events, and the capability to incorporate both facts and previous data.

Random Forest (RF) Algorithm

These algorithms represent different random elements to create different decision factories in sets. If there are classification problems, the results of these trees are summarised for the final forecast. When creating ensemble classifiers, randomization plays a significant role in creating a wide range of models based on deterministic algorithms. Using integrated methods, several models are combined to improve the generalizability of the resulting classifiers. Traditionally, aggregation methods relied on deterministic algorithms with randomized process to generate various options.

Representatives of deterministic algorithms are Bagging, RF, Randomized C4.5 and Random Subspace. Individual DT and correlation among base trees are key issues that decide the RF classifier's performance. Because of this, the proposed enhanced random forest optimizes a large number of decision trees by selecting only uncorrelated data and good trees with high classification accuracies. The tree selection process has the following steps:

1. Identifying and selecting only the good trees with high classification accuracy.
2. The correlation is measured between the selected good tees.
3. Based on the measured correlation, only uncorrelated trees are selected.

RF algorithm is a popular technique for building ML systems. This is a supervised ML method proposed by Leo Bremen. RF is an integrated algorithm. A set consists of separately trained algorithms called core algorithms, whose predictions are combined to predict new events. RF uses the decision tree as the main algorithm. Generate multiple decision trees and combine the results of these decision trees as a final decision. RF introduces randomness in two ways:

1. Bootstrap samples are generated by drawing random samples from a dataset.
2. Random selection of attributes or input features for producing separate base decision trees.

When RF is used for classification, the results of the basic decision trees are combined with a majority data to obtain better results.

Multi-Layer Perceptron (MLP)

MLP is the most widespread neural network structure, especially the two-layered structures where the input blocks and output layers are connected to hidden layers in between. Each neuron model in the network contains a non-linear activation function that is different. As a result, it can perform a static association between the network input area and the output space. On the other hand, MLPs often have connections from hidden neurons to a layer of reference units with a time delay. These blocks store the output of hidden neurons (including 1 weight) for a one-time pass and then return it to the input level. In this way, the hidden neurons record their previous activity, which allows the network to perform incremental learning tasks over time.

The MLP is assumed to provide a non-linear mapping among the input vector and the consistent output vector. A large part of the work in this field has been dedicated to maintain this non-linear mapping in a static context. Several attempts have made to expand the MLP architecture in order to understand the category of problems. In the case of a feedback network or a repeating network, for example, the prior state of the network can be sent back to the input. An important advantage of the multi-layer perceptron is that the coefficients can be easily adjusted by using a method that has been successful in practice and is known as the regeneration algorithm. It is used to describe neural networks. It is a supervised learning method in which the network's output is compared to the signal required during the training phase to determine how well it works. A reprocessing algorithm is a type of algorithm of sharper descent in which an error signal shows the variance between the current output of the neural network and the anticipated output is used to adjust the weights in the output layer and then used for weight measurement. Next, calculate and adjust the inputs in hidden lines back over the network. While the neural network processes the input signals for output at full power, the resulting error multiplies from the output to the network during training to adjust the weight.

AN OVERVIEW OF ML’S SECURITY

In this section, we provide an overview of the safety of ML, particularly from a IDS perspective, and highlight the various safety challenges associated with the use of ML.

Security Threats

In the threats of security, ML can be categorized into three dimensions, namely, impact attacks, security breaches, and attack details [35]. Figure 1 shows a classification of these threats to the security of ML systems.

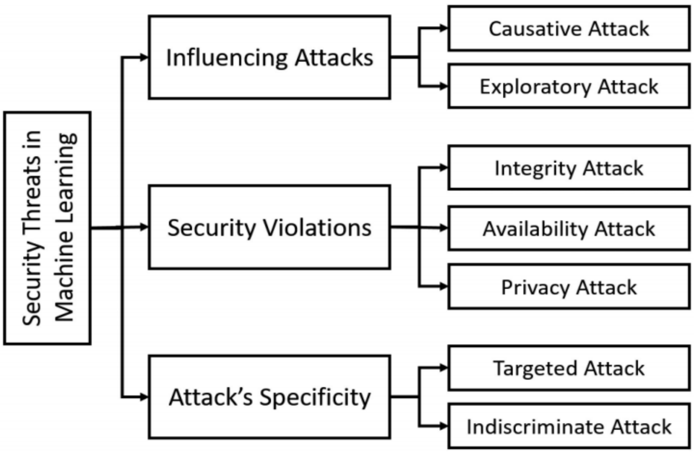

Fig 1 Classification of Security Threats

a) Impact: Impact attacks can be of two types: (1) Causal: the type of training that attempts to control the data; (2) Exploration: exploits the error classification of the ML model without interfering with the model's training.

b) security breach: refers to the availability and integrity of the service, which can be categorized into three types: (1) integrity attack: attempts to increase the false negative rate of the executing model (attacker) when the model provides malicious input; (2) Availability attack: unlike security attack, it seeks to increase the classifier's false positive rate in response to inappropriate input; (3) Privacy attack: Refers to the disclosure of sensitive and confidential information from training data or from a trainer model, or both.

c) The specificity of the attack: the specificity of the attack can be determined in two ways: (1) the target attack: whether the attack is in a specific input sample or a set of samples; (2) Random attack: Causes a random failure of the LA model.

The first axis of attack classification in ML systems (shown in Figure 1) determines the capabilities of the enemy, for example, whether the training process can be modified by injecting toxic data (ie trying to access the training data). If the attacker does not have access to the training data, the attacker can launch a heuristic attack, for example, considering a disease classification problem, exploiting the question-response pairs to achieve the desired behavior of the opponent (i.e. classification is incorrect in this case). The second dimension of the attacks relates to the type of security breaches that an adversary can commit, for example, trying to learn the privacy of users in training data or trying to increase the rate of false negatives or false positives. from the workbook. Each type of security breach is a major issue for healthcare applications, which means that maintaining user privacy is a major concern, and low-resolution models are highly desirable. The third dimension describes the specific objectives of the opponent. An attacker might want to test the target attack, for example, forcing a classifier to classify a particular input sample into a target class (for example, by trying to bypass the detector to discover that the input is not malicious) or radically cracking the classifier.

Adversarial ML

Aggressive attacks are the result of recent attempts to identify training and inference weaknesses in MLmodels. Hostile attacks have become the biggest security threat to ML systems [36] - [39]. The adversarial examples are generated by introducing small unnoticeable perturbation into non-modified samples to erradicate the integrity of ML system. The next sub-section will descibe the two different tyoes of adversarial attacks.

a) Toxic/Poisoning attacks: The model training is affected by this adversarial attacks. In other words, the learning of ML model is mislead by manipulating the training data is called as poisoning attacks [40].

b) Evacuation attacks: The inference phase of the training process is affected by this attack and it is known as evasion attack [41]. The integrity of the ML model is compromised by the manipulating the test data in this attacks and therefore attacker provides harmful to the inputs.

RELATED WORKS OF NIDS IN CLOUD, MANET AND WANET, WSN AND IOT NETWORKS

Huge networks like cloud environments and large IT ecosystems need a collaborative and extremely well-organized technique for attaining their security objectives. For the protection of such networks, collaborative IDS(CIDS) have emerged to detect sophisticated and highly distributed attacks [42]. MANETs have inherent vulnerability features like open medium, highly dynamic network topology, limited physical security, lack of centralized monitoring and control system, etc. [43]. Vehicular ad hoc network (VANET) technology is grabbing the attention of all modern transportation systems. This technology is also vulnerable to different kinds of attacks [44].

VANET nodes can share their experiences and thus they can improve attack detection accuracy. Distributed ML is an appropriate structure for the implementation of this kind of cooperative attack/anomaly detection over VANETs. This Collaborative learning is also prone to attack as a malicious node can infer sensitive information from the data shared by other nodes in the network. The privacy-preserving ML-based collaborative IDS (PML-CIDS) algorithm can be used as a classifier to detect the intrusion type [45]. This algorithm’s privacy notation is captured by differential privacy methods. Here, training data privacy protection and optimized security and privacy in VANET are the main objectives. NSL-KDD dataset is used in this work.

Bigdata techniques are also adopted in VANETs for handling a huge volume of data. Spark-ML RF-Based detection algorithm is suggested for DDoS attack detection in Bigdata generated from VANET [46]. A Micro-batch data processing technique is used for network traffic collection and feature extraction. Experiments are conducted on NSL-KDD and UNSW-NB15 datasets. Better accuracy and false positive rate (FPR) are the achievements of this experiment. The author suggested the deployment of the proposed NIDS in a real environment as future work.

Momani et al [47] generates a new IDS dataset (WSN-DS) for WSN using an NS2 network simulator with five different states: normal, black hole attack, flood attack, table attack, and gray hole attack. The authors used ANN to detect attacks on WSN-DS. Otoum et al [48] proposed a cluster-based IDS model for WSN. In this model, intrusion detection in CH was performed using two subsystems: RF, Enhanced Density Dependent Noise Applications, Spectral Clustering (E-DBSCAN). RF is used to detect known attacks and E-DBSCAN is used to detect unknown attacks.

Otoum et al. [49] are comparing IDS based on MLand IDS based on deep learning of WSN. The authors determined that IDS based on deep learning provides higher accuracy compared to IDS based on ML, but IDS based on deep learning takes longer time to detect attacks compared to IDS based on ML. Most researchers use the KDD dataset to test the IDS model offline for WSN. But the KDD dataset is a class unbalanced dataset. Due to the unbalanced data set, no accurate results are obtained. Tan and others. [50] Use the SMOTE algorithm to perform class imbalance and then use the random forest algorithm to perform intrusion detection on the KDDCup'99 dataset.

To detect the various attacks, author from [51] proposed a centralized approach using ANN. The weight is optimized by two algorithms called grey wolf and evolutionary system in [51]. In WSN, cyber attack is detected by distributed approach in Betam et al [52]. The normal or abnormal traffic is identified by ant colony and particle swarm optimization with high classification accuracy in Nithyanandam et al. [53]. This paper simulation is carried out by NS-2 in WSN.

Three algorithms such as cultural algorithm, adaboost and artifical fish swarm algorithm for IDS in WSN by the author from [54]. The misuse detection is performed by the dataset called NSL-KDD. Genetic algorithm is proposed by Singh et al. [55] to perform energy efficient IDS. Four modules were used in this work. In IoT, the wormhole attack is detected by Pongle and Chavan [56] and calculated the attacker loaction. Power consumption is low, because this method is light-weighted.

The comparison of different algorithms for effective IDS in MANET is provided by Pastrana et al [57]. From this review, the author concludes two algorithms such as SVM and GA are better, because low overhead is obtained by them. A survey of IDS in IoT domain is conducted by author from [58] in 2017. The future scope and issues of IDS in IoT are explained in detail. In order to protect the data in IoT domain, NB is proposed by Ahmed and others [59]. The discovery rate is high and load on distributed network is reduced by this model. A node is classified by binary logistic regression in Yuano et al. [60]. The attacks of black hole and selective redirection are monitored by the local sensors using this method.

PROBLEMS ASSOCIATED WITH NIDS

The higher false detection rate of IDS

Anomaly-Based NIDS wrongly categorizes normal but previously unseen system activities as an anomaly. This increases the FP rate. On the other hand, the reason behind the increase in the false-negative rate is the high frequency of new attacks introduced in cyberspace nowadays. Signature-based NIDS stores known attack signatures, and cannot detect new attacks [61-62]. Moreover, some signature based NIDS may be so specific that a mild change in attack can avoid its detection. In such situations, security experts have no awareness that an attack took place. False negatives cannot be easily judged. Theoretically, a blend of signature-based detection and anomaly based detection approaches is supposed to be an improvement over both the single approaches. But in the case of a hybrid approach where an anomaly detector creates a list of anomalous observations that are further classified by a signature-based detector into known attacks. In such a case, if the anomaly detector fails to detect an attack because of its similarity with normal behavior patterns, it cannot be detected by the signature-based detector in a later stage [63].

IDS evaluation with large real-time network traffic

Drastically increased internet data and users making it a puzzling task to monitor huge real-time network traffic. It is essential for the improvement of IDSs to thoroughly analyze both normal as well as abnormal traffic behavior. Learning and detecting attack patterns accurately from such huge data requires a huge amount of training data for generating better results. Modeling and evaluating IDSs with large real network traffic is one of the current key challenges.

Inefficient intrusion datasets

Enormous unknown patterns of network intrusions are detected recently which are still growing in count continuously at a rapid rate. Therefore updating intrusion datasets periodically is a necessity. This will help in representing appropriate architecture for testing old as well as recently observed network anomalies. A big concern in a multi-cloud environment is the unavailability of datasets for recent security attack analysis, due to privacy issues [64]. The unavailability of efficient intrusion datasets comprising an adequate amount of relevant intrusion types is a big issue.

Data imbalance

Uneven record distribution in imbalanced Datasets leads to the biased classification of records. The detection rate of a class with fewer records is very less as compared to the detection rate of a class with a majority of records [65]. A variety of data balancing techniques are available to convey this issue but at the cost of increased computational complexity and execution time complexity.

Slow learner IDS

Slow learner IDS is another issue usually ignored. But it should be dealt with with proper attention to fulfill the need for current requirements of the big data situation [66]. Timely detection of intrusion can save target systems/organizations from massive damage.

Tedious class labeling in supervised learning IDS approaches. Class labeling is very tedious for field experts when a dataset reaches multi-gigabytes or even more in size

IDS protection

Besides IDS by itself is prone to be attacked . ML-based IDS models learn attack patterns from the input data. Such models can be a victim of the adversarial attack, wherein minor modification can disguise the traffic classifier. Watermarking techniques are gaining popularity in protecting software against cyber-attacks [67]. Such techniques can be utilized for IDS protection.

CONCLUSION

This paper focuses on important aspects in the area of network intrusion detection. Many NIDS are built using publically available intrusion datasets. This paper highlights the characteristics and limitations of a variety of publicly available intrusion datasets including the Botnet dataset and Malware datasets. This has resulted in a better understanding of the nature and area of applications of these datasets. It also concludes that there is a need to update intrusion datasets and generate new comprehensive, and efficient datasets. Important aspects of ML techniques are discussed with their application on intrusion detection. ML techniques are competent to handle a large amount of evolving and complex data, but, these techniques have their characteristics and limitations that are to be considered before building a NIDS model. This paper also presents a study of recent NIDS models that exploited the ML-techniques and public intrusion datasets. Different networking environments are considered to conduct this survey. This study presents a clear vision of the current security challenges, solutions, outcomes, and future directions. Hence, this work will be helpful to the researchers to identify a suitable dataset and ML techniques for effective IDS modeling in different networking environments for carrying out their research. Further, this paper can be extended to analyze the real-time monitoring of rapidly increasing network traffic which is still a challenge and an interesting topic of current network security researches.

REFERENCES

1. M. Kemiche and R. Beghdad, Intelligent Systems in Science and Information 2014: Extended and Selected Results from the Science and Information Conference 2014, Cham: Springer International Publishing, ch. Towards Using Games Theory to Detect New U2R Attacks, pp. 351–367, (2015).
2. S. Patil, D. V. K. B. P, S. Singha and R. Jamil, A Survey on Authentication Techniques for Wireless Sensor Networks, International Journal of Applied Engineering Research, vol. 7, (2012).
3. T. M. Mitchell, Machine Learning, 1st ed, New York, NY, USA: McGraw-Hill, Inc., (1997).
4. G. Poojitha, K. N. Kumar and P. J. Reddy, Intrusion Detection Using Artificial Neural Network, In 2010 International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–7, July (2010).
5. H. Altwaijry and S. Algarny, Bayesian Based Intrusion Detection System, Journal of King Saud University – Computer and Information Sciences, vol. 24, no. 1, pp. 1–6, (2012). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319157811000292
6. M. Panda and M. R. Patra, Semi-naive Bayesian Method for Network Intrusion Detection System, In Neural Information Processing, 16th International Conference, ICONIP 2009, Bangkok, Thailand,

December 1–5, 2009, Proceedings, Part I, pp. 614–621, (2009). [Online]. Available: http://dx.doi.org/10.1007/978-3-642-10677-4-70

7. M. Panda, A. Abraham and M. R. Patra, A Hybrid Intelligent Approach for Network Intrusion Detection, Procedia Engineering, vol. 30, pp. 1–9, (2012), International Conference on Communication Technology and System Design 2011. [Online]..
8. Ahmed M, Mahmood AN, Hu J. A survey of network anomaly detection techniques. J Netw Comput Appl. 2016;60:19–31.
9. Ring M, Wunderlich S, Scheuring D, et al. A survey of network-based intrusion detection data sets. Comput Secur. 2019;86:147–167.
10. Divekar A, Parekh M, Savla V, et al. Benchmarking datasets for anomaly based network intrusion detection: KDD CUP 99 alternatives. 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS); 2018 October; IEEE. p. 1–8.
11. Lippmann R, Haines JW, Fried DJ, et al. The 1999 DARPA off-line intrusion detection evaluation. Comput Netw. 2000;34(4):579–595.
12. Elkan C. Results of the KDD'99 classifier learning contest. Sponsored by the International Conference on Knowledge Discovery in Databases; 1999 September.
13. Engen V, Vincent J, Phalp K. Exploring discrepancies in findings obtained with the KDD Cup'99 data set. Intell Data Anal. 2011;15(2):251–276.
14. Tavallaee M, Bagheri E, Lu W, et al. A detailed analysis of the KDD CUP 99 data set. 2009 IEEE symposium on computational intelligence for security and defense applications; 2009 July; IEEE. p. 1–6.
15. McHugh J. 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 (TISSEC). 2000;3(4):262–294.
16. Devan P, Khare N. An efficient XGBoost–DNN-based classification model for network intrusion detection system. Neural Comput Appl. 2020;32:1–16.
17. Khare N, Devan P, Chowdhary CL, et al. SMO-DNN: spider monkey optimization and deep neural network hybrid classifier model for intrusion detection. Electronics(Basel).2020;9(4).
18. Kolias C, Kambourakis G, Stavrou A, et al. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Commun Surv Tutorials. 2015;18(1):184–208.
19. Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Military communications and information systems conference (MilCIS), 2015. New York: IEEE; 2015. p. 1–6.
20. Moustafa N, Slay J. The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf Secur J: A Global Perspect. 2016;25(1-3):18–31.
21. Koroniotis N, Moustafa N, Sitnikova E, et al. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Future Gener Comput Syst. 2019;100:779–796.
22. Taheri R, Ghahramani M, Javidan R, et al. Similarity-based Android malware detection using hamming distance of static binary features. Future Gener Comput Syst. 2020;105:230–247.
23. Zhou Y, Jiang X. Dissecting android malware: characterization and evolution. 2012 IEEE symposium on security and privacy; 2012, May; IEEE. p. 95–109.
24. Arp D, Spreitzenbarth M, Hubner M, et al. Drebin: effective and explainable detection of android malware in your pocket. In: Ndss. Vol. 14.. San Diego: Internet Society; 2014 Feb. p. 23–26.
25. Contagio Dataset. 2020. [ctied 2020 Oct 21] Available from: http://contagio minidump.blogspot.com/. https://www.sec.cs.tu-bs.de/ ~ danarp/drebin/.
26. Ghahramani Z. Unsupervised learning. In: Bousquet O, Von Luxburg, Rätsch G, editors. Summer school on machine learning. Berlin, Heidelberg: Springer; 2003 Feb. p. 72–112.
27. S. S. Liu, L. F. Zhang, Z. Yan, Predict pairwise trust based on machine learning in online social networks: a survey, IEEE Access, 6 (1) (2018) 51297-51318.

28. L. F. Wei, W. Q. Luo, J. Weng, Y. J. Zhong, X. Q. Zhang, and Z. Yan, Machine learning-based malicious application detection of android, IEEE Access, 5 (1) (2017) 25591-25601.
29. H. Q. Lin, G. Liu, Z. Yan, Detection of application-layer tunnels with rules and machine learning, The 12th International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage (SpaCCS2019), 2019, pp. 441-455.
30. A. L. Samuel, Some studies in machine learning using the game of checkers. I, Computer Games I, (1988) 335–365.
31. X. Jing, Z. Yan, and P. Witold, security data collection and data analytics in the Internet: a survey. IEEE Communications Surveys & Tutorials, 21 (1) (2018) 586-618.
32. T. Kanungo, D. M. Mount, et al., An efficient k-means clustering algorithm: analysis and
33. implementation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (7) (2002) 0-892.
34. J. Matousek, On approximate geometric k-clustering, Discrete & Computational Geometry, 24 (1) (2000) 61-84.
35. Kamarularifin Abd Jalil, Muhammad Hilmi Kamarudin, Mohamad Noorman Masrek "Comparison of Machine Learning Algorithms Performance in Detecting Network Intrusion", 2010.
36. M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," inProc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur., 2015, pp. 1322–1333.
37. I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in Proc. Int. Conf. Learn. Representat., 2015.
38. N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in Proc. IEEE Eur. Symp. Secur. Privacy, 2016, pp. 372–387.
39. N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in Proc. ACM Asia Conf. Comput. Commun. Secur., 2017, pp. 506–519.
40. M. Usama, J. Qadir, A. Al-Fuqaha, and M. Hamdi, "The adversarial machine learning conundrum: Can the insecurity of ML become the achilles' heel of cognitive networks?" IEEE Netw., vol.34, no. 1, pp. 196–203, 2019.
41. B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in Proc. 29th Int. Conf. Mach. Learn., 2012, pp. 1807–1814.
42. B. Biggio et al., "Evasion attacks against machine learning at test time," in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases, 2013, pp. 387–402
43. Vasilomanolakis E, Karuppayah S, Mühlhäuser M, et al. Taxonomy and survey of collaborative intrusion detection. ACM Comput Surv (CSUR). 2015;47(4):1–33.
44. Djenouri D, Khelladi L, Badache AN. A survey of security issues in mobile ad hoc and sensor networks. In: Dusit Niyato, editor. IEEE Communications surveys Tutorials. Vol. 7, No. 4Singapore: IEEE Communications Society; 2005. p. 2–28.
45. Pathan ASK, ed. Security of self-organizing networks: MANET, WSN, WMN, VANET. CRC press; 2016.
46. Zhang T, Zhu Q. Distributed privacy-preserving collaborative intrusion detection systems for VANETs. IEEE Trans Signal Inf Process Over Netws. 2018;4(1):148–161.
47. Gao Y, Wu H, Song B, et al. A distributed network intrusion detection system for distributed Denial of service attacks in vehicular Ad Hoc network. IEEE Access. 2019;7:154560–154571.
48. I. Almomani, Bassam Al-Kasasbeh and Mousa AL-Akhras, "WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks," Journal of Sensors, Vol. 2016, 16 pages, 2016.
49. S. Otoum, B. Kantarci and H. T. Mouftah, "Detection of Known and Unknown Intrusive Sensor Behavior in Critical Applications," IEEE Sensors Letters, Vol. 1, No. 5, pp. 1-4, Oct. 2017
50. S. Otoum, B. Kantarci and H. T. Mouftah, "On the Feasibility of Deep Learning in Sensor Network Intrusion Detection," IEEE Networking Letters, 2019.
51. X. Tan, S. Su, Z. Huang, X. Guo, Z. Zuo, X. Sun and L. Li, "Wireless Sensor Networks Intrusion Detection Based on SMOTE and the Random Forest Algorithm," Sensors, Vol. 19, No. 203, p. 203, 2019.

52. A. Mansouri, B. Majidi and A. Shamisa, ”Metaheuristic Neural Networks for Anomaly Recognition in Industrial Sensor Networks with Packet Latency And Jitter for Smart Infrastructures,” International Journal of Computers and Applications, 2018.
53. S. Bitam, S. Zeadally and A. Mellouk, "Bio-Inspired Cybersecurity for Wireless Sensor Networks," IEEE Communications Magazine, Vol. 54, No. 6, pp. 68-74, 2016.
54. N. Nithiyanandam, P. Latha Parthiban, B. Rajalingam, “Effectively Suppress the Attack of Sinkhole in Wireless Sensor Network using Enhanced Particle Swarm Optimization Technique,” International Journal of Pure and Applied Mathematics, Vol. 118, No. 9, pp. 313-329, 2018.
55. X. Sun, B. Yan, X. Zhang, and C. Rong, “An Integrated Intrusion Detection Model of Cluster-based Wireless Sensor Network,” Plos One, Vol. 10, No. 10, 2015.
56. S. Singh and R. S. Kushwah, “Energy Efficient Approach for Intrusion Detection System for WSN by Applying Optimal Clustering and Genetic Algorithm,” In Proceedings of the Int. Conf. on advances in info. Commu. tech. & comput.—AICTC ’16, pp. 1–6, New York, 2016.
57. Pongle, P., & Chavan, G. (2015). Real time intrusion and wormhole attack detection in Internet of Things. International Journal of Computer Application, 121(9), 1–9.
58. Pastrana, S., Mitrokotsa, A., Orfla, A., & Peris-Lopez, P. (2012). Evaluation of classifcation algorithms for intrusion detection in MANETs. Knowledge-Based Systems, 36, 217–225.
59. Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in internet of things. Journal of Network and Computer Applications. https://doi. org/10.1016/j.jnca.2017.02.009.
60. Mehmood, A., Mukherjee, M., & Ahmed, S. A. (2018). NBC-MAIDS: Naïve Bayesian classifcation technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks. The Journal of Supercomputing, 74(10), 5156.
61. Ioannou, C., Vassiliou, V., & Sergiou, C. (2017). An intrusion detection system for wireless sensor networks. In 24th international conference on telecommunications (ICT) (pp. 3–5).
62. Hajisalem V, Babaie S. A hybrid intrusion detection system based onABC-AFS algorithm for misuse and anomaly detection. Comput Netw.2018;136:37–50.
63. Lee W, Stolfo SJ, Mok KW. A data mining framework for building intrusiondetection models. Proceedings of the 1999 IEEE Symposium on Security andPrivacy (Cat. No. 99CB36344); 1999, May; IEEE. p. 120–132.
64. Tombini E, Debar H, Mé L, et al. A serial combination of anomaly and misuseIDSes applied to HTTP traffic. 20th Annual Computer Security ApplicationsConference; 2004 December; IEEE. p. 428–437.
65. Salman T, Bhamare D, Erbad A, et al. Machine learning for anomaly detection and categorization in multi-cloud environments. 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud); 2017,June. IEEE. p. 97–103.
66. Patel H, Singh Rajput D, Thippa Reddy G, et al. A review on classificationof imbalanced data for wireless sensor networks. Int J Distrib Sens Netw.2020;16(4):155014772091640.
67. Tan Z, Nagar UT, He X, et al. Enhancing big data security with collaborativeintrusion detection. IEEE Cloud Computing. 2014;1(3):27–33.
68. [67]. Iwendi C, Jalil Z, Javed AR, et al. Keysplitwatermark: zero watermarking algorithm for software protection against cyber-attacks. IEEE Access.2020;8:72650–72660

How to cite this article:

T S Ramya, N Subha Raagavi, Dr N Karpagavallii, “Future Trends in Network Based Intrusion Detection Systems”, International Journal of Intelligent Computing and Technology (IJICT), Vol.5, Iss.2, pp.39-54, 2022