IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

MACHINE LEARNING TECHNIQUES FOR INTRUSION DETECTION IN NETWORKS:A SYSTEMETIC STUDY

[1]Nidhi Nigam,[2] Nisha Rathi,[3]Shivshankar Rajput

[1]Assitant Professor, [2]Assitant Professor,[3]Assitant Professor

[1] Department of Computer Science and Engineering,

[1]Acropolis Institute of Technology and Research, Indore, India

Abstract: Intrusion detection system is one of the implemented solutions against harmful attacks. Moreover, attackers try to always change their tools and techniques. Although, implementing an accepted IDS system is additionally a difficult task. In this paper, several methods are identified and reviewed to assess various machine learning algorithms. Additionally, several conditions for wireless communication for deciding whether to apply ML as well suitable technique of ML. Also traditional summarized approaches along with their performance comparison with ML based techniques are surveyed.

Index Terms – Intrusion Detection System (IDS), Machine Learning Algorithms.

I. INTRODUCTION

Over the past few years Machine learning becomes an emerging as well as a central, typically hidden part of life. Applications and advancements of Machine Learning are everywhere in the world, one can see. Undoubtedly ML has been applied to various complex types of problems arising in various domains. This paper address the use of ML approaches in networks operation and its maintenance. Machine Learning applications are important from home appliances to autonomous vehicles. Furthermore it will become an increasingly important factor of human life, which aids decision making, analysis and automation.

Since network operations majorly demands efficient solutions, so it is require to use powerful ML techniques for higher network performance. The association of network design with machine learning provides the possibility to generate new network applications. As machine learning compromised supervised and unsupervised training which includes classification and regression as example of supervised data, which was used as traditional ML technique in networks from so many years. The magical power of Machine Learning can be seen from so many recent developments from infrastructure to distributed data processing frameworks.

Machine Learning for Networking is suitable and efficient for following reasons. Primarily classification and prediction plays important role in detection of intrusions and performance prediction [1]. Additionally, it helps in decision making which is helpful in network scheduling [2] and according to current status of parameter adaptation [3,4]. Machine learning can provide an estimated model of these systems with acceptable accuracy. Finally, each network environment may have different characteristic (e.g., traffic patterns and network states) and researchers often need to solve the problem for each scenario independently. Machine learning may provide new possibilities to construct the generalized model via a uniform training method [3, 4].

II. LITERATURE REVIEW

To protect networks against cyber threats [5] that may compromise the networks availability, or yield unauthorized access or misuse of network resources comes under networks security. Undoubtedly business processes are directly or indirectly changing which not only cost billions of dollars in damage and recovery [6]. Therefore security of networks is fundamental in network operations and management.

Nowadays attackers are constantly trying to fetch the data by any method so this becomes very challenging for security experts to develop measures to secure network from attacks and make zero-day attacks. Some of security measures are to apply encryption techniques on networks parameters like network traffic, payload, integrity etc. Additionally authorization using credentials, access control, anti viruses and firewalls are another parameters that affects the security.

However, encryption keys and login credentials can be breached, exposing the network to all kinds of threats. Furthermore, the prevention capabilities of firewalls and anti-viruses are limited by the prescribed set of rules and patches. Hence, it is imperative to include a second line of defense that can detect early symptoms of cyber-threats and react quickly enough before any damage is done. Such systems are commonly referred to as Intrusion Detection/Prevention Systems (IDS/IPS). IDSs monitor the network for signs of malicious activities and can be broadly classified into two categories—Misuse- and Anomaly-based systems. While the former rely on signatures of known attacks, the latter is based on the notion that intrusions exhibit a behavior that is quite distinctive from normal network behavior. Hence, the general objective of anomaly-based IDSs is to define the "normal behavior" in order to detect deviations from this norm.

When it comes to the application of ML for network security, through our literature survey we have found that the majority of works have focused on the application of ML for intrusion detection. Here, intrusion detection refers to detecting any form of attacks that may compromise the network e.g. probing, phishing, DoS, DDoS, etc. This can be seen as a classification problem. While there is a body of work on host-based intrusion detection (e.g. malware and botnet detection), we do not delve into this topic, as most of these works utilize traces collected from the end-host (sometimes in correlation with network traces). Concretely, in our discussion, we focus on network-based intrusion detection and we classify the works into three categories, namely misuse, anomaly, and hybrid network IDSs.

III. MISUSE-BASED INTRUSION DETECTION

Misuse-based IDSs consist of monitoring the network and matching the network activities against the expected behavior of an attack. The key component of such a system is the comprehensiveness of the attack signatures. Typically, the signatures fed to misuse-IDS rely on expert knowledge [7]. The source of this knowledge can either be human experts, or it can be extracted from data. However, the huge volume of generated network traces renders manual inspection practically impossible. Furthermore, attack signatures extracted by sequentially scanning network traces will fail to capture advanced persistent threats or complex attacks with intermittent symptoms. Intruders can easily evade detection if the signatures rely on a stream of suspicious activities by simply inserting noise in the data.

In light of the above, ML became the tool of choice for misuse-based IDSs. Its ability to find patterns in big datasets fits the need to learn signatures of attacks from collected network traces. Hence, it comes as no surprise to see a fair amount of literature [7, 8,9,10,11,12] that rely on ML for misuse-detection. Naturally, all existing works employ supervised learning, and the majority performs the detection offline. Note, we classify all work that use normal and attack data in their training set as misuse-detection.

The earliest work that employed ML for misuse detection is [7]. It was among the first to highlight the limitations of rule-based expert systems, namely that they (i) fail to detect variants of known attacks, (ii) require constant updating, and (iii) fail to correlate between multiple individual instances of suspicious activities if they occur in isolation. Following the success of NN in the detection of computer viruses, the application of NN for misuse detection as an alternative to rule-based systems is proposed. The advantages of NN are its ability to analyze network traces in a less structured-manner (as opposed to rule-based systems), and to provide prediction in the form of a probability. The latter can enable the detection of variants of known attacks. For evaluation, training and testing dataset are generated using *RealSecure* ^{TM}—a tool that monitors network data and compares it against signatures of known attacks. For attack dataset, *InternetScanner* ^{TM} [13] and Satan Scanner [14] tools are used to generate port scans and syn-flood attacks on the monitored host. Results show that the NN is able to correctly identify normal and attack records 89-91% of the time.

Amor et al. [8] compare NB and DT also using KDD'99 dataset, and promote NB's linear training and classification times as a competitive alternative to DT. NB is found to be 7 times faster in learning and classification than DT. For whole attacks, DT shows a slightly higher accuracy over NB. However, NB achieves better accuracy for DoS, R2L, and probing attacks. Both NB and DT perform poorly for R2L and U2R attacks. In fact, Sabhnani and Serpen [398] expose that no classifiers can be trained successfully on the KDD dataset to perform misuse detection for U2R or R2L attack categories. This is due to the deficiencies and limitations of the KDD dataset rather than the inadequacies of the proposed algorithms.

The authors found via multiple analysis techniques that the training and testing datasets represent dissimilar hypothesis for the U2R and R2L attack categories; so if one would employ any algorithm that attempts to learn the signature of these attacks using the training dataset is bound to perform poorly on the testing dataset. Yet, the work in [12] reports surprisingly impressive detection accuracy for U2R and R2L. Here, a hybrid of BP NN with C4.5 is proposed, where BP NN is used to detect DoS and probing attacks, and C4.5 for U2R and R2L. For U2R and R2L only a subcategory of attacks is considered (yielding a total of 11 U2R connections out of more than 200 in the original dataset and ~2000 out of more than 15000 for R2L connections). After-the-event analysis is also performed to feed C4.5 with new rules in the event of misclassification.

Other seminal works consider hybrid and ensemble methods for misuse detection [16, 17, 18]. The goal of ensemble methods is to integrate different ML techniques to leverage their benefits and overcome their individual limitations. When applied to misuse detection, and more specifically for the KDD'99 dataset, these work focused on looking at which ML technique works best for a class of connections. For instance, Peddabachigari et al. [17] propose an IDS that leverages an ensemble of DT, SVM with polynomial kernel based function, and hybrid DT-SVM to detect various different cases of misuse. Through empirical evaluation, the resultant IDS consist of using DT for U2R, SVM for DoS, and and DT-SVM to detect normal traffic. The ensemble of the 3 methods together (with a voting mechanism) is used to detect probing and R2L attacks. The resultant accuracy for each class is presented in Table 1.

Stein et al. [18] employ DT with GA. The goal of GA is to pick the best feature set out of the 41 features provided in KDD'99 dataset. DT with GA is performed for every category of attacks, rendering a total of 4 DTs. The average error rate achieved by each DT at the end of 20 runs is reported in Table 21. Another interesting ensemble learning approach is the one proposed in [90], where the ensemble is composed of pairs of feature set and classification technique. More specifically, BN and CART classification techniques are evaluated on the KDD'99 dataset with different feature sets. Markov blanket [353] and Gini [76] are adopted as feature selection techniques for BN and CART, respectively. Markov blanket identifies the only knowledge needed to predict the behavior of a particular node; a node here refers to the different categories of attacks. Gini coefficient measures how well the splitting rules in CART separates between the different categories of attacks. This is achieved by pruning away branches with high classification error. For BN, 17 features out of 41 are chosen during the data reduction phase. For CART, 12 variables are selected. CART and BN are trained on the 12 and 17 features set, as well as 19 features set from [326]. They describe the final ensemble method using pairs (#features, classification), which delineates the reduced feature set and the classification technique that exhibits the highest accuracy for the different categories of attacks and normal traffic. The ensemble model achieves 100% accuracy for normal (12 features set, CART), probe (17 features set, CART), and DoS (17 features set, Ensemble), and 84% accuracy for U2R (19 features set, CART), and 99.47% accuracy for R2L (12 features set, Ensemble).

Miller et al. [22] also devise an ensemble method but based on NB classifiers, denoted as Multi-perspective Machine Learning (MPML). The key idea behind MPML is that an attack can be detected by looking at different network characteristics or "perspective". These characteristics in turn are represented by a subset of network features. Hence, they group the features of a perspective together, and train a classifier using each feature set. The intuition behind this approach is to consider a diverse and rich set of network characteristics (each represented by a classifier), to enhance the overall prediction accuracy. The predictions made by each classifier are then fed to another NB model to reach a consensus.

A limitation of the aforementioned approaches is that they are all employed offline, which inhibits their application in real life. A few related works focused on the training and detection times of their IDS. Most classifiers (e.g., image, text recognition systems) require re-training from time to time. However, for IDSs this retraining may be performed daily (or even hourly) due to the fast and ever changing nature of cyber-threats [180]. Hence, fast training times are critical for an adaptable and robust IDS. [198] tackled the challenge of devising an IDS with fast training time using an Adaboost algorithm. The proposed algorithm consists of an ensemble of weak classifiers (decision stumps), where their decisions are then fed to a strong classifier to make the final decision. The fast training time achieved (of 73 s) is attributed to the use of weak classifiers. Another advantage of decision stumps is the ability to combine weak classifiers for categorical features with weak classifiers for continuous features, without any forced conversation as is typically done in most works. During the evaluation, a subset of attack types are omitted from the training set in order to evaluate the algorithm's ability to detect unknown attacks. While the reported accuracy is not significantly high (90%), the training time is promising for real-time deployment. Clearly, there is still a need for a model that can achieve fast training time, without sacrificing the detection accuracy.

Sangkatsanee et al. [402] propose real-time misuse-based IDS. Information gain is applied to reduce the number of features used (for faster detection), resulting in 12 features. Different ML techniques were assessed, among which DT provided the best empirical results. They developed a tool that runs on traces collected in 2 s time intervals, and shows a detection accuracy of 98%. A post-processing technique is also proposed to reduce FP, which consists of flagging an attack only if 3-out-of 5 consecutive records belonging to the same connection were classified as an attack. While this work is indeed promising, given it is performed in real-time, it suffers from a few limitations: (i) it can only detect two types of attacks (DoS and probe), (ii) it is not compared against other real-time signature-based IDS (e.g. Snort [87]), (iii) it only looks at attacks in windows of 2 s, and (iv) its post-processing approach correlates records between 2 IPs, making it vulnerable to persistent threats and distributed attacks.

A final effort that merits a discussion here is [272]. This work employs Transductive Confidence Machine for k-NN (TCM-KNN), a supervised classification algorithm with a strangeness measure. A high strangeness measure indicates that the given instance is an outlier in a particular class (for which the measurement is being conducted). The strangeness measure is calculated for every instance against each possible classification class. This is achieved by measuring the ratio of the sum of the k-nearest distances from a given class to the sum of the k-nearest distances from all other classes. The strangeness measure is also employed for active learning. Since getting labeled data for attacks is a cumbersome task, active learning can relieve part of this tedious process by indicating the subset of data points that should be labeled to improve the confidence of the classifier. TCM-KNN is evaluated over the KDD'99 dataset and the results are reported in Table 21. The benefit of active learning is also evaluated. Starting with a training set of just 12 instances, TCM-KNN requires the labeling of an additional 40 actively selected instances to reach a TP of 99.7%. Whereas, random sampling requires the labeling of 2000 instances to attain the same accuracy.

TABLE 1

SUMMARY OF ML-BASED MISUSE DETECTION

REF.	ML TECHNIQUE	DATASET	FEATURES	EVALUATION	
				SETTINGS	RESULTS
Cannady [7]	Supervised NN (offline)	Normal: RealSecure Attack: [13,14]	TCP, IP, and ICMP header fields and payload	-1 Layer MLP: 9, [a], 2 -Sigmoid function -Number of nodes in hidden layers determined by trial & error	DR: 89%-91% Training + Testing runtime: 26.13 hrs
Pan et al. [15]	Supervised NN and C4.5 DT (offline)	KDD Cup	all 41 features	-29,313 training data records -111,858 testing data records -1 Layer MLP: 70-14-6 -NN trained until MSE = 0.001 or # Epochs = 1500 -Selected attacks for U2L and R2L -After-the-event analysis	DR Normal : 99.5% DR DoS: 97.3% DR Probe (Satan): 95.3% DR Probe (Portsweep): 94.9% DR U2R: 72.7% DR R2L: 100% ADR: 93.28% FP: 0.2%
Moradi et al. [11]	Supervised NN (offline)	KDD Cup	35 features	-12,159 training data records -900 validation data records -6,996 testing data records -Attacks: SYN Flood and Satan -2 Layers MLP: 35 35 35 3 -1 Layer MLP: 35 45 35 -ESVM Method	2 Layers MLP DR: 80% 2 Layers MLP Training time > 25 hrs 2 Layers MLP w/ ESVM DR: 90% 2 Layers MLP w/ ESVM Training time < 5 hrs 1 Layers MLP w/ ESVM DR: 87%
Chebrolu et al. [16]	Supervised BN and CART (offline)	KDD Cup	Feature Selection using Markov Blanket and Gini rule	-5,092 training data records -6,890 testing data records - AMD Athlon 1.67 GHz processor with 992 MB of RAM	DR Normal: 100% DR Probe: 100% DR DoS: 100% DR U2R: 84% DR R2L: 99.47% Training BN time: 11.03 ~ 25.19 sec Testing BN time: 5.01 ~ 12.13 sec Training CART time : 0.59 ~ 1.15 sec Testing CART time: 0.02 ~ 0.13 sec

Amor et al. [8]	Supervised NB (offline)	KDD Cup	all 41 features	-494,019 training data records -311,029 testing data records -Pentium III 700 Mhz processor	DR Normal: 97.68% PCC DoS: 96.65% PCC R2L: 8.66% PCC U2R: 11.84% PCC Probing: 88.33%
Stein et al. [18]	Supervised C4.5 DT (offline)	KDD Cup	GA-based feature selection	-489,843 training data records -311,029 testing data records -10-fold cross validation -GA ran for 100 generations	Error rate DoS: 2.22% Error rate Probe: 1.67% Error rate R2L: 19.9% Error rate U2R: 0.1%
Miller et al. [22]	Supervised Ensemble MPML (Offline)	NSL-KDD	all 41 features	-125,973 training records -22,544 testing records -3 NBs trained w/ 12, 9, 9 features -Platform used Weka	TP: 84.137% FP: 15.863%
Li et al. [23]	Supervised TCM K-NN (Offline)	KDD Cup	all 41 features 8 features selected using Chi-square	-Intel Pentium 4, 1.73 GHz, 1 GB RAM, Windows XP Professional -Platform Weka -49,402 training records -12,350 testing records -K = 50	41 features: TP 99.7% 41 features: FP 0% 8 features: TP 99.6% 8 features: FP 0.1%

[A]Determined empirically, Mean Square Error (MSE), Percentage Correct Classification (PCC), Average Detection Rate (ADR), Early Stop Validation Method (ESVM)

IV. CONCLUSION

To conclude, most works on the application of ML for intrusion detection are offline, and amongst the few real-time IDSs, there is no consideration for early detection (i.e. detecting a threat from the first few packets of a flow). Moreover, there is a gap in the ML for intrusion detection literature with regards to intrusion detection for persistent threats, or correlating among isolated anomaly instances over time. Finally, only a handful of works have actually evaluated the robustness of their algorithm in the event of mimicry attacks, an aspect of critical importance as attackers are constantly looking for ways to evade detection.

.

References

[1] Y. Sun et al., "CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction," Proc. SIGCOMM 2016, ACM, pp. 272–85.

[2] B. Mao et al., "Routing or Computing? The Paradigm Shift Towards Intelligent Computer Network Packet Transmission Based on Deep Learning," IEEE Trans. Computers, 2017.

[3] K. Winstein and H. Balakrishnan, "TCP Ex Machina: Computer-Generated Congestion Control," Proc. ACM SIGCOMM Computer Commun. Rev., vol. 43, no. 4, ACM, 2013, pp. 123–34.

[4] M. Dong et al., "PCC: Re-Architecting Congestion Control for Consistent High Performance," Proc. NSDI 2015, pp. 395–408.

[5] Lab Kaspersky. Damage control: The cost of security breaches. IT security risks special report series Report, Kaspersky. 2015. https://media.kaspersky.com/pdf/it-risks-survey-report-cost-of-security-breaches.pdf. Accessed 10 Nov 2017.

[6] Juniper Research. Cybercrime will cost businesses over $2 trillion by 2019. 2015. https://www.juniperresearch.com/press/press-releases/cybercrime-cost-businesses-over-2trillion. Accessed 10 Nov 2017.

[7] Cannady J. Artificial neural networks for misuse detection. In: Proceedings of the 21st National information systems security conference, vol. 26. Virginia: 1998. p. 368–81.Google Scholar

[8] Amor NB, Benferhat S, Elouedi Z. Naive bayes vs decision trees in intrusion detection systems. In: Proceedings of the 2004 ACM symposium on Applied computing. ACM: 2004. p. 420–4.Google Scholar

[9] Chebrolu S, Abraham A, Thomas JP. Feature deduction and ensemble design of intrusion detection systems. Comput secur. 2005; 24(4):295–307.CrossRefGoogle Scholar

[10] Kruegel C, Toth T. Using decision trees to improve signature-based intrusion detection. In: Recent Advances in Intrusion Detection. Springer: 2003. p. 173–91.Google Scholar

[11]Moradi M, Zulkernine M. A neural network based system for intrusion detection and classification of attacks. In: Proceedings of the IEEE International Conference on Advances in Intelligent Systems-Theory and Applications.2004. p. 15–8.Google Scholar

[12]Pan ZS, Chen SC, Hu GB, Zhang DQ. Hybrid neural network and c4.5 for misuse detection.2003, pp. 2463–7.Google Scholar

[13]Proactcouk. ISS Internet Scanner. 2017. http://www.tech.proact.co.uk/iss/iss_system_scanner.htm. Accessed 28 Dec 2017.

[14]Farmer D, Venema W. Satan: Security administrator tool for analyzing networks. 1993. http://www.porcupine.org/satan/. Accessed 28 Dec 2017.

[15]Pan ZS, Chen SC, Hu GB, Zhang DQ. Hybrid neural network and c4.5 for misuse detection.2003, pp. 2463–7.Google Scholar

[16]Chebrolu S, Abraham A, Thomas JP. Feature deduction and ensemble design of intrusion detection systems. Comput secur. 2005; 24(4):295–307.CrossRefGoogle Scholar

[17]Peddabachigari S, Abraham A, Grosan C, Thomas J. Modeling intrusion detection system using hybrid intelligent systems. J netw comput appl. 2007; 30(1):114–32.CrossRefGoogle Scholar

[18]Stein G, Chen B, Wu AS, Hua KA. Decision tree classifier for network intrusion detection with ga-based feature selection. In: Proceedings of the 43rd annual Southeast regional conference-Volume 2. ACM: 2005. p. 136–41.Google Scholar

[19]Breiman L, Friedman J, Stone C, Olshen R. Classification and Regression Trees. The Wadsworth and Brooks-Cole statistics-probability series. New York: Taylor & Francis; 1984.Google Scholar

[20]Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference.Morgan Kaufmann; 2014.Google Scholar

[21]Mukkamala S, Sung AH, Abraham A. Intrusion detection using ensemble of soft computing paradigms. In: Intelligent systems design and applications. Springer: 2003. p. 239–48.Google Scholar

[22]Miller ST, Busby-Earle C. Multi-perspective machine learning a classifier ensemble method for intrusion detection. In: Proceedings of the 2017 International Conference on Machine Learning and Soft Computing. ACM: 2017. p. 7–12.Google Scholar

[23]Li Y, Guo L. An active learning based tcm-knn algorithm for supervised network intrusion detection. Comput Secur. 2007; 26(7):459–67.CrossRefGoogle Scholar