Babylonian Journal of Machine Learning
Vol.2023, **pp**. 26–30
DOI: https://doi.org/10.58496/BJML/2023/005; ISSN: xxxxx
https://mesopotamian.press/journals/index.php/BJML

Review Article

Reviews research on applying machine learning techniques to reduce false positives for network intrusion detection systems

Kavita Rajora [1],* , Nazar salih abdulhussein [2],

[1]*Agra College, Agra, India*
[2] *Imam aladham University College, Iraq*

ARTICLE INFO

Article History
Received 04 Jan 2023
Accepted 09 Mar 2023
Published 28 May 2023

Keywords
Machine Learning
IDS
ELM
HMM
Anomaly Detection

ABSTRACT

High false positive rates impede the adoption of anomaly detection methods, which have promise for detecting novel cyber threats. Techniques reviewed include Extreme Learning Machine (ELM), Hidden Markov Models (HMM), situation awareness frameworks, ensemble methods, and feature selection algorithms when applied to contemporary benchmark datasets. Findings show combinations of ELM, HMMs, and ensemble classifiers can achieve reduced false positive rates. However, gaps still exist in research using current representative data.

1. INTRODUCTION

Intrusion Detection Systems (IDSs) are hardware and software tools that detect, analyze, and report intrusions in computer systems and networks. They can be used in-line to detect and prevent intrusions in near real-time, or communicate with other security devices like firewalls to automatically implement blocking rules in response to detection. Research into IDSs began several decades ago with Anderson's paper on computer threat monitoring and surveillance. IDSs are characterized as either misuse detection based on known attack patterns or signatures, or anomaly detection based on deviations in behavior from normal. Misuse detection is preferred in commercial environments due to higher accuracy, while anomaly detection is preferred in academic research due to its potential to recognize novel attacks. Recent research has used Extreme Learning Machine (ELM) and Hidden Markov Models (HMMs) to improve intrusion detection. The concept of situation awareness, first introduced by Endsley in 1988, has been applied to various domains, including speech and handwriting recognition[1].

Existing anomaly detection techniques for Internet of Things (IoT) have a high False Positive Rate (FPR), which is a significant issue in IDS effectiveness. Anomaly-based methods, which show promise in detecting novel cyber-attacks, generate more false positives than signature-based methods. The growth in network sizes and increasing complexity and variability of attacks contribute to this uncertainty. Axelsson (2000) argued that the false alarm rate should be measured in relation to how many intrusions one would expect to detect rather than the maximum number of possible false alarms. Perdisci, Ariu, Fogla, Giacinto, and Lee (2009) concluded that FPRs for IDSs must be very low. Sommer and Paxson (2010) concurred that reducing false positives in anomaly detection for IDS must be a top priority due to the high operational cost of IDS error rates. This research aimed to develop a new approach for anomaly-based IDS that can better minimize false positives while detecting indications of contemporary cyber-attacks without sacrificing accuracy[2].

2. REVIEW OF THE LITERATURE

This review aims to understand the issue of reducing false positives in intrusion detection by examining various areas of the literature. It begins with an overview of Intrusion Detection Systems (IDSs), their history, evolution post-Denning, and

*Corresponding author. Email: kavitarajora2@gmail.com

various types of IDSs. The review then delves into how IDSs are evaluated, including measuring false positives (FPs) and other metrics. It also discusses popular data sets for training and testing, their challenges, and the techniques used in machine learning. The review also discusses situation awareness and different types of cyber-attacks, often illustrated using the UNSW-NB15 dataset for training and testing[3].

2.1 Misuse vs Anomaly Detection

Misuse detection (also called signature-based detection) works by matching events/traffic against predefined attack patterns or rules. Requires knowledge of attacks.

Anomaly detection works by developing a model of normal behavior/traffic and identifying deviations from that model as potential attacks. No prior attack knowledge needed[4].

2.2 Signature-based vs Machine Learning Systems

Signature-based systems like Snort rely on manmade rules defining malicious traffic patterns. Limited against new attacks.

Machine learning systems build statistical models to classify traffic as normal or attack. Can detect novel attacks but often have higher false positives[5].

2.3 Host-based vs Network-based IDS

Host-based IDSs (HIDS) analyze system logs on an individual computer/host for signs of intrusion.

Network-based IDSs (NIDS) analyze network traffic data looking for attacks across multiple hosts[6].

2.4 Key Performance Measurements

Accuracy: Percentage of correct classifications of normal and attack traffic

$$Accuracy = TP + TN / TP + FP + TN + FN$$

False Positive Rate (FPR): Percentage of normal traffic incorrectly classified as attacks

$$False\ Positive\ Rate\ (FPR) = FP/ FP + TN$$

A False Negative (FN) occurs when the predicted output is negative, or normal for this research, but the actual instance is positive.

$$False\ Negative\ Rate\ (FNR) = FN /FN + TP$$

Tradeoff between high DR and low FPR - tuning threshold affects both[7].

3. FEATURE SELECTION

The literature review discusses the importance of feature selection in classifier design, focusing on removing redundant and irrelevant features to reduce computational complexity and algorithm costs. Two feature selection studies were reviewed for UNSW-NB15 data, with the first using a hybrid technique based on Central Points and Associate Rule Mining to reduce the number of features to 11. The second study specifically used the Train and Test data set, using Information Gain and CFS algorithms to validate the Moustafa and Slay (2017) study and compare it to KDD99[8].

In this research, appropriate features for both the ELM and HMM classifiers were evaluated and chosen using Information Gain and CFS, respectively. Information Gain was chosen for the ELM model as it is an independent filter method that allows the same attributes to be used for other techniques. A cutoff or threshold point needed to be determined on where to draw the line to select the number of attributes. The cut-off point was determined using an informed estimate based on other feature selection studies using UNSW-NB15, and an additional experiment was conducted using backwards feature removal to determine a cut-off point[9].

For the HMM models, CFS was used to choose a subset of features in contrast to the ELM model, which uses Information Gain. CFS is a multivariate method that is created based on the fact that good feature subsets are highly correlated with the class yet uncorrelated to each other. For each feature selected by CFS, an HMM was created, creating a series of classifiers which were then combined using a voting scheme[10].

The ELM classifier is focused on a 64-point time point, while the HMM classifiers are focused on temporal patterns. The selection of CFS provides for classifiers that are highly correlated to the class but uncorrelated to each other, resulting in improved classifier accuracy[11].

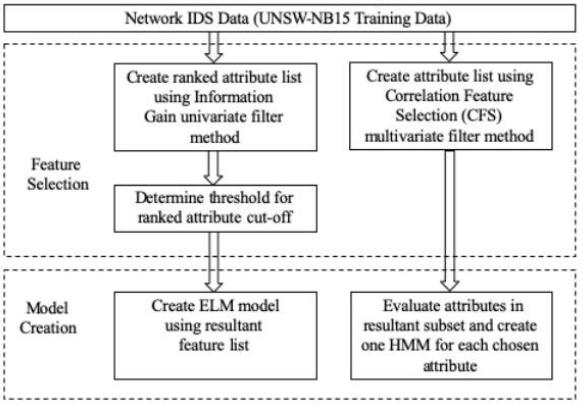

Fig.1. Feature Selection Process

4. CLASSIFIER

The ELM classifier was developed to classify normal and attack traffic using the UNSW-NB15 data sets. The basic ELM program, MATLAB version, was obtained to implement the ELM classifier. The required inputs for the program include file names for training and testing data sets, ELM type (regression or classification), the number of hidden neurons, and the activation function. Supported activation functions include sigmoidal, sine, hardlim, triangular basis, and radial basis[12].

The preprocessing for each data set used for an ELM run included moving the binary label for attack and normal traffic to the first column, removing the label for attack category, converting nominal or categorical attributes to binary attributes, and normalizing all numeric attributes between -1 and 1. The resulting test and training data sets were saved as tab-delimited text files[12].

The sigmoid activation function was chosen based on past experience, and the ELM Type value was set to 1 for classification. The program also requires the number of hidden neurons as input. Determining the number of hidden neurons is an open problem for ELM researchers, but Huang, Zhu, and Siew (2006) demonstrated that ELM is very stable across a wide range of hidden nodes but performance can degrade with too few or too many nodes[13].

An HMM architecture requires the specification of five parameters: N (the number of states) and M (the number of distinct observation symbols per state) and three probability measures: A (the state transition probability distribution), B (the observation symbol emission probability distribution), and p (the initial state distribution). Various values of N (states) have been used for intrusion detection in the literature[13].

For each HMM classifier, symbols were selected using the letters of the alphabet to represent each possible feature value or range of values for the feature. Each symbol was mapped to its state, either normal, probe, or attack, for each record in the

training data. The probabilities were determined through training using the Baum-Welch algorithm, which is the most commonly used HMM training algorithm[14].

The HMMs were implemented using the machine learning tool kit in MATLAB, with training performed using the hmmestimate and hmmtrain functions, and testing performed with the hmmviterbi function.

The combined classifier was the combined output of the HMMs and ELM, providing both a projection in time and space. A unanimous voting scheme was chosen for the final output, with the goal of eliminating false positives and ensuring all classifiers agree on an attack. More complicated voting schemes were discounted due to prior research showing only a marginal difference in performance between simple voting schemes and more advanced combination techniques[15].

5. DISCUSSION

The research focused on the evaluation of FPR using a situation awareness framework. Experiment A achieved a 10% FPR reduction, under 0.6% FPR, and better FPR than a J48 DT or MLP classifier using 18 and 24 attributes. However, the difference between the two models and 60 and 125 hidden neurons was not significant.

Experiment B achieved a 10% FPR reduction and under 0.6% FPR, with the J48 DT achieving a better FPR than the MLP classifier. However, both solutions had an FPR of 0%, resulting in a tie for the MLP classifier. This resulted in a slight degraded accuracy rate.

In experiment B, two of the three HMMs had perfect and almost perfect TPR but poor FPR performance. However, due to the unanimous voting scheme, the high rates of false positives were cancelled out for the overall ensemble.

6. CONCLISION

The combined classifier was the result of combining the outputs of the Hidden Markov Models (HMMs) and Extreme Learning Machines (ELM), which offered both temporal and spatial projection. The final output was determined using a unanimous vote mechanism, aiming to eliminate false positives and ensure consensus among all classifiers regarding an assault. Advanced voting systems were disregarded since previous research shown only a minimal disparity in effectiveness between simple voting schemes and more sophisticated combination techniques.

Conflicts Of Interest

The author's paper declares that there are no relationships or affiliations that could create conflicts of interest.

Funding

The acknowledgments section of the paper does not mention any financial support from institutions or sponsors.

Acknowledgment

The author acknowledges the support and resources provided by the institution in facilitating the execution of this study.

References

[1] Al-Yaseen, W.L., Othman, Z.A., & Nazri, M.Z.A. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. Expert Systems with Applications, 67, 296-303.

[2] Fiore, U., Palmieri, F., Castiglione, A., & De Santis, A. (2017). Network anomaly detection with the restricted Boltzmann machine. Neurocomputing, 122, 13-23.

[3] Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A deep learning approach for network intrusion detection system. Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (pp. 21-26). ICST.

[4] Kevric, J., Jukic, S., & Subasi, A. (2017). An effective combining classifier approach using tree algorithms for network intrusion detection. Neural Computing and Applications, 28(S1), 1051-1058.

[5] Li, W., Yi, P., Wu, L., Pan, L., & Wu, J. (2017). A new intrusion detection system based on KNN classification algorithm in wireless sensor network. Journal of Electrical and Computer Engineering, 2017.
[6] Muna, A.H., Moustafa, N., & Sitnikova, E. (2018). Identification of malicious activities in industrial internet of things based on deep learning models. Journal of Information Security and Applications, 41, 1-11.
[7] Shone, N., Ngoc, T.N., Phai, V.D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41-50.
[8] Sindhu, S.S.S., Geetha, S., & Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach. Expert Systems with Applications, 39(1), 129-141.
[9] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525-41550.
[10] Wang, G., Hao, J., Ma, J., & Huang, L. (2010). A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. Expert Systems with Applications, 37(9), 6225-6232.
[11] Xiao, L., Chen, Y., & Chang, X. (2014). Bayesian model averaging of Bayesian network classifiers for intrusion detection. Proceedings of the 8th International Symposium on Computational Intelligence and Design, 1, 451-454.
[12] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5, 21954-21961.
[13] Zong, B., Huang, G., & Chen, H. (2013). Weighted extreme learning machine for imbalance learning. Neurocomputing, 101, 229-242.
[14] Zuech, R., Khoshgoftaar, T.M., & Wald, R. (2015). Intrusion detection and Big Heterogeneous Data: a survey. Journal of Big Data, 2(3).
[15] Wang, J., Huang, H., Sun, J., & Sun, J. (2019). Squeeze Extreme Learning Machine for intrusion detection. IEEE Access, 7, 107955-107962.