A MACHINE LEARNING APPROACH FOR INTRUSION DETECTION FOR NETWORK DATASET

1st Yogesh Thakre
Computer Department at Sinhgad college of Engineering Savitribai Phule Pune University
Pune, India
yogesht@7302@gmail.com
2nd Aditya Ghadge
Computer Department at Sinhgad college of Engineering Savitribai Phule Pune University
Pune, India
aditya.b.ghadge3177@gmail.com

3rd Vitthal Kale
Computer Department at Sinhgad college of Engineering Savitribai Phule Pune University
Pune, India
vitthalkale9359@gmail.com
4th Harshal Doshi
Computer Department at Sinhgad college of Engineering Savitribai Phule Pune University
Pune, India
harshaldoshi18@gmail.com
5th Prof. Laxman Pawar
dept. name of organization (of Aff.)
Computer Department at Sinhgad college of Engineering
Pune, Maharashtra
lbpawar.scoe@sinhgad.edu

Abstract

The research paper proposes an intrusion detection method called Incremental Learning and FSVMIL-FSVM, aim-ing to address the limitations of traditional network intrusion detection algorithms in terms of high learning time cost and low recognition accuracy for large-scale training data. The field of networking has experienced significant growth in recent decades, leading to increased threats to computer networks from attackers and hackers. To detect such attacks, an Intrusion Detection System (IDS) is used. The paper introduces a new algorithm, Genetic Algorithm using machine learning, to improve accuracy and speed in detecting network intrusions. The rapid growth of network data and the emergence of various intrusion types highlight the need for effective intrusion detection methods.

Index Terms—*Intrusion detection, Pattern Based Intrusion Detection, Intrusion Detection using Statistics.*

I. INTRODUCTION

The research paper focuses on the implementation of in-trusion detection systems (IDS) using data mining and deep learning approaches for cyber security. The objective is to detect new attacks, improve accuracy compared to exist-ing approaches, and increase the detection rate. The paper highlights the importance of IDS as part of the system's defence line against cyber-attacks and emphasizes the need for higher accuracy and robust behaviour in the face of modern sophisticated threats. The motivation for developing IDS is to identify and distinguish malicious activities or intrusion attempts, which can compromise the integrity, privacy, and accessibility of a system. An IDS is a software system that monitors and reports on such activities. The paper recognizes the growing trend of utilizing computer-based techniques for fault identification and emphasizes the importance of quick detection and high accuracy in building a responsive system.

The literature survey section briefly mentions two rele-vant papers. The first paper proposes distributed IDS using blockchain and cloud computing infrastructure, highlighting the need for efficient utilization of resources and leveraging modern techniques. The second paper discusses a pattern-based intrusion detection model that combines statistical-based and pattern-based approaches for improved intrusion detection. The integration of these approaches offers a comprehensive system to detect intrusions.

In summary, the research paper aims to address the chal-lenges in intrusion detection by leveraging data mining and deep learning approaches. It emphasizes the need for accurate and robust IDS systems to counter modern cyber-attacks and highlights previous studies that propose innovative techniques such as distributed IDS and pattern-based intrusion detection.

II. RELATED WORK

The related work on intrusion detection systems (IDS) em-phasizes key aspects such as external interface requirements, non-functional requirements, and software quality attributes. IDSs are deployed alongside access control, authentication mechanisms, and encryption techniques to enhance system security. The hardware requirements include 8GB RAM, an Intel i5 processor, and the Spyder IDE for efficient data processing. The software interface comprises Windows 10 as the operating system, Spyder as the IDE, and Python as the programming language.

Non-functional requirements focus on performance and safety. The IDS should exhibit fast performance in functions, encryption, and virtual environment provision. The modular design enables easy error detection, fixing, and installation of new functionality. Safety requirements ensure easy error detection and fixing, as well as seamless installation and updates.

The software quality attributes include adaptability, avail-ability, maintainability, reliability, user friendliness, integrity, and security. Adaptability caters to users with different system configurations, while availability ensures easy access for all users. Maintainability allows for easy error resolution, and reliability is crucial for the software's performance. User friendliness is achieved through a GUI, and integrity controls unauthorized access. Security measures, including multi-phase user authentication, provide robust protection against intru-sions.

This summary highlights the importance of considering external interface requirements, non-functional requirements, and software quality attributes in IDS implementation. These factors contribute to the effectiveness of intrusion detection systems in securing systems against cyber threats.

III. IMPLEMETATION

The first step in intrusion detection is the collection of relevant data from various sources within the network. This includes network traffic data, system logs, user activity logs, and other relevant information. The data can be collected through network taps, sensors, logging mechanisms, or by using specialized software agents installed on network devices.

Once the data is collected, it needs to be preprocessed to remove noise, irrelevant information, and to transform it into a suitable format for analysis. Preprocessing techniques may include data cleaning, normalization, feature extraction, and dimensionality reduction. This step aims to enhance the quality of the data and optimize it for further analysis. Intru-sion detection systems employ various techniques to analyze network traffic and identify potential threats. This includes both signature-based and anomaly-based analysis.

Signature-based analysis involves comparing the collected data against a database of known attack signatures or patterns. If a match is found, it indicates the presence of a known attack or intrusion. Signature-based analysis is effective in detecting known attacks but may fail to identify new or unknown threats.

Anomaly-based analysis focuses on identifying deviations from normal network behavior. It establishes a baseline of normal network activity and flags any activities that devi-ate significantly from this baseline as potential intrusions. Anomaly detection algorithms can use statistical analysis, ma-chine learning, or artificial intelligence techniques to identify anomalous patterns or behaviors.

Intrusion detection systems can also utilize rule-based detec-tion techniques. In this approach, predefined rules or patterns are created based on known attack behaviors. These rules

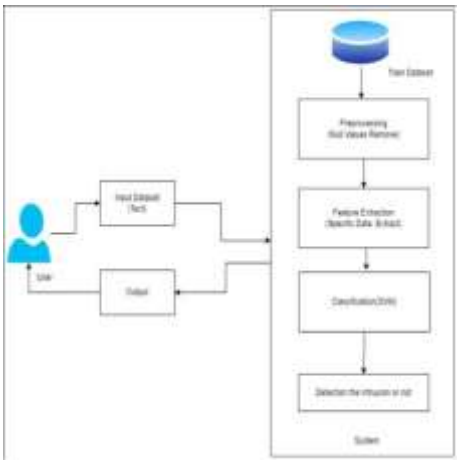

Fig. 1. System architecture

define specific conditions or events that, when detected in network traffic or system logs, indicate a potential intrusion. Rule-based detection can be effective for detecting known at-tack patterns but may generate false positives or miss unknown attacks.

When an intrusion is detected, an appropriate response and mitigation strategy need to be implemented. This may involve generating alerts or notifications to system administrators, blocking suspicious network traffic, isolating compromised systems, or initiating automated responses to mitigate the impact of the intrusion. The response strategy depends on the severity and nature of the detected intrusion.

Intrusion detection is an ongoing process that requires continuous monitoring and updates. New threats and attack techniques emerge regularly, and IDS systems need to be regularly updated with the latest threat intelligence and detec-tion algorithms. This includes updating the signature database, retraining machine learning models, and implementing new rules or policies to adapt to changing threat landscapes.

The performance of intrusion detection systems needs to be evaluated periodically to assess their effectiveness and identify areas for improvement. Evaluation metrics such as detection rate, false positive rate, and response.

IV. TESTING

Software testing is an essential part of the development process and can be implemented at different stages depending on the chosen methodology. In traditional models, most testing occurs after requirements have been defined and coding is complete. Newer models, like Agile, often involve test-driven development, with developers taking on a larger role in testing before formal testing teams get involved. There are two main testing strategies: black-box testing and white-box testing. Black-box testing focuses on examining the functionality of the software without knowledge of its internal implementation. Testers are only aware of what the software is supposed to do. On the other hand, white-box testing takes an internal perspective and uses programming skills to design test cases.

Different types of testing are used in software development. Unit testing involves testing individual software units after completion but before integration. It validates the internal program logic, ensuring that inputs produce valid outputs. Integration testing checks if integrated software components function as one program, exposing problems that may arise from their combination. System testing involves sequencing various types of testing, such as unit, integration, validation, GUI, and low and high-level test cases. It focuses on testing the application's functionality, data transmission, and overall performance.

In summary, software testing is crucial for ensuring the quality and functionality of software applications. It can be conducted at different stages of development, depending on the chosen methodology. The testing strategies include black-box and white-box testing, and various types of testing, such as unit, integration, and system testing, are employed to validate different aspects of the software's behaviour and performance.

V. CONCLUSION AND FUTURE SCOPE

In conclusion, this research paper explores deep learning approaches for intrusion detection, focusing on deep dis-criminative models and generative/unsupervised models. The effectiveness of these methods is evaluated by comparing them using two new datasets and specifically analyzing Support Vector Machines (SVM). As a future scope, the study aims to investigate methods for developing dynamic models that eliminate the need for retraining, thereby reducing costs as-sociated with changes in intra-site cost metrics and enabling deployment in diverse sites with varying cost models. These advancements hold the potential to enhance the efficiency and cost-effectiveness of intrusion detection systems, paving the way for improved cybersecurity measures in the future.

ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to ev-eryone who contributed to the completion of this research paper. Firstly, we are deeply thankful to our guide [Prof. Laxman Pawar] for her invaluable guidance and support. We also extend our appreciation to the members of our research committee for their insightful feedback.

REFERENCES

[1] (2018), "Evaluation of Academic Performance Based on Learning An-alytics and Ontology: a Systematic Mapping Study," IEEE ; San Jose, USA, pp. 1-5. DOI: 10.1109/FIE.2018.8658936 .

[2] "Conversational Agents in ELearning," in Applications and Innova-tions in Intelligent Systems XVI, London: Springer London, 2009, pp. 169–182.

[3] , "Bots as language learning tools," Lang. Learn. Technol., vol. 10, no. 3, pp. 8–14, 2006.

[4] Current Development of Adaptive E-learning systemHambleton, Ad-vances in assessment models, methods, and practices. New York, USA: American Council on Education/Macmillan, 1996.

[5] Almond, "Bayes Nets in Educational Assessment : Where Do the Numbers Come From ? CSE Technical Report 518 Duanli Yan , and Linda S . Steinberg CRESST / Educational Testing Service March 2000 Center for the Study of Evaluation National Center for Research on Evalua," vol. 1522, no. 310, 2000.

[6] Web 3.0 in Education Research BIJIT - BVICAM's International Journal of Information Technology Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM)New Delhi Copy Right © BIJIT – 2011; July – December, 2011; Vol. 3 No. 2; ISSN 0973 – 5658 335

[7] A Survey of Object-Oriented Petri Nets and Analysis Methods. IEICE Trans.Fundamentals, Vol. E88–A, No.11 (2005)

[8] A Survey of Object-Oriented Petri Nets and Analysis Methods. IEICE Trans.Fundamentals, Vol. E88–A, No.11 (2005)

[9]] Al-Janabi, S.T.F.; Saeed, H.A., "A Neural NetworkBased Anomaly Intrusion Detection System," Developments in E-systems engineer-ing (DeSE), 2011, vol., no., pp.221, 226, 6-8 Dec. 2011 DOI: 10.1109/DeSE.2011.1

[10] V Kosamkar, SS Chaudhari, Improved Intrusion Detection System using C4.5 Decision Tree and Support Vector Machine, International Journal of Computer Science and Information Technologies, 2014, 5 (2), 1463-1467.

[11] KVR Swamy, KSV Lakshmi, Network Intrusion Detection Using Im-proved Decision Tree Algorithm, International Journal of Computer Science and Information Security, 2012, 3 (5), 26-32.

[12] MS Hoque, M Mukit, M Bikas, A Naser, An Implementation of Intrusion Detection System using Genetic Algorithm, International Journal of Network Security and its Applications, 2012, 4(2), 109-120.

[13] Y. Yi, J. Wu, and W. Xu, "Incremental SVM set for network intrusion detection," Ex Applications, vol. 38, pp. 7698-7707, 6// 201

[14] .F.Lin, S.D.Wang. Fuzzy support vector machines. IEEE Transactions On Neural Network, 2002, 13(3): 466-471.

[15] LIU Ye, WANG Zebing, FENG Yan, GU Hongying. DoS Intrusion Detection Based on Incremental Learning with Support Vector Ma-chines.Computer Engineering, 2006, 32(4),pp: 179-180, 186.

[16] Wang Sheng, Jin Zhigang. IDS classification algorithm based on fuzzy SVM models. Application Research of Computers,2018, Vol. 37 No. 2.

[17] Tan Ai-ping, Chen Hao,Wu Bo-qiao.Network Intrusion Intelligent De-tection Algorithm Based on AdaBoost. Computer Science, 2014, Vol.41, No.2,pp:197-120.

[18] YAO Wei,WANG Juan ,ZHANG Shengli. Intrusion detection model based on decision tree and Naive-Bayes classification. Journal of Com-puter Applications, 2015, Vol.35,No. 10, pp:2883-2885