International Journal on Robotics, Automation and Sciences

Performance Improvement Scheme of NIDS Through Optimized Intrusion Pattern Database

Nouman Amjad, Anam Mumtaz, Sara Abbas, and Umar Hayat*

***Abstract* – Network-based intrusion detection systems (NIDS) are perceptively distributed devices within computer networks. They aim to examine traffic passing through the network on which they are installed passively. The database is the most vital part of network intrusion detection systems, as all the data converted information from the NIDS needs to be saved in a patterned structured manner. Understanding the usability of several available types of databases like central databases, Distributed databases, operational databases, etc., it is on the developer's end to choose the most comprehensive one. Data transformation and performance speed are essential features that a stable database can handle. In this paper, we have analyzed the performance of multiple databases to find out the proficient way that favors NIDS optimization.**

Keywords— Distributed Database, Classification Algorithms, Network Intrusion Detection System, Anomaly Detection, Pattern Matching

I. INTRODUCTION

The storage systems are on the frontier for protecting against intrusions. As data storage could see variations in persistently stored data, multiple types of intrusions might be triggered along with data management. Organizations are financing in information technology and IT cyber defense competencies to guard their valuable assets. An enterprise's necessity is to protect customer information and its intellectual capital; the attack detection as well as response to protect interests of organizations which have three attributes in common: processes, people, and technology. So, the data management and retrieval process must be very important for each organization. System storage is important to every domain because it compromises highly proficient methods for handling multiple data. In computer network security, the choice of data storage within system compatibility is highly intensive.

Due to the tremendous rise of technologies daily, the complications of data management and safety measures increased. It is significant to accomplish data resourcefully and permit the network system to execute tasks easily [1]. Data protection of computer networks is a vital task for developers. Data protection procedures are deployed over Intrusion detection systems to make IDS data more protected from malicious attacks and

*Corresponding Author email: uhayat.buic@bahria.edu.pk
Nouman Amjad, is with Curtin University Western Australis (email: itsnoumanamjad@yahoo.com)
Anam Mumtaz, is with Computer Science department, Bahria University, Pakistan (email: anammumtaz325@gmail.com)
Sara Abbas was with Computer science department, Bahria University, Pakistan (email: saraabbas00011@gmail.com)
*Umar Hayat was with Computer science department, Bahria University, Pakistan.(email: uhayat.buic@bahria.edu.pk)

International Journal on Robotics, Automation and Sciences (2024) 6,1:64-69
https://doi.org/10.33093/ijoras.2024.6.1.9
Manuscript received: 21 Dec 2023 | Accepted: 26 Mar 2024 | Published: : 30 Apr 2024
Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

several vulnerabilities. To overcome security and low-performance issues, IDS is coupled with different schemes, including Signature matching based and anomaly detection based schemes.

Most researchers express their work towards making the system efficient and accurate, but these collective efforts are mostly to detect IDS or OS-level operations. Unfortunately, they cannot retrieve corrupt data due to unethical database transactions. Data corruption spreads throughout the database and could multiply its damaging effect speedily [2]. This is the most vital danger for database applications around the globe. These persistent attacks on databases are required to be detected accurately over time. If not, it would be easier for administrators to recover the damages soon. Computer storage comprises mainly two types: primary storage and secondary storage. System performance is dependent upon its data-dealing procedures [3].

For high performance, most data is stored at the system's primary storage; in most cases, it may be random access memory RAM. Primary storage is volatile memory. While executing an application, the most nearby proficient source of data retrieval is RAM. It is used in such types of system applications, in which applications' main concern is with performance nether with stored data. Secondary storages refer to such storage devices that are not consistently accessible by the computer system; this includes portable use, hard drives etc. Such appliances are either inserted or plugged in order to access the system. Secondary storage is nonvolatile data information because it is stored in a computer's hard drive computer. That software application uses secondary storage, which is concerned with stored data. Most applications use secondary storage for reuse or presenting the data as full information. Later on, the stored data and processed information are used to develop a "System performance visualization tool". The graphical visuals could help non-technical staff understand system performance so that non-technical members could easily operate the system.

II. Database Valuations Over NIDS

Intrusion Detection Systems (IDS) typically utilize secondary storage for storing data and logs. Signature-based IDSs operate on a predefined set of rules to detect and prevent cyber-attacks on the network. [4], [6]. These rules are matched among incoming and outgoing network traffic. If the traffic stream matches the predefined ruleset, an alert is generated by IDS toward the network administrator. Intrusion detection system IDS reads and understands rules from its database.

Pre-determined Signatures are already available to IDS. After pattern matching each signature, intrusion detection system IDS produces a comprehensive number of system Alerts and Logs [7], [8]. These logs and alerts are stored in the database because they are useful to system administrators. IDS data storage is not just for logs storage purposes, but we need to develop a wide-ranging database management system that's empowered with data organizational measures [9]. For rapid processing, the stored data is required to be in structured form. This will help the system to process data retrieval commands quickly [10]. Beyond fundamentals of storage policies, CRC-NIDS desires to make a strategic function for its own pre-defined rules management [11]. IDS produces information logs for different multi-sources, which is obligatory for the IDS database management system (DBMS) to control log flow toward organized data storage hubs. Research was conducted to evaluate diverse database systems. The exploration and classification of databases are detailed in Figure 1.

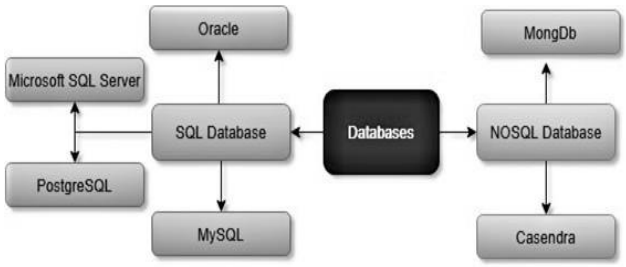

FIGURE 1. In-depth exploration and classification of databases.

III. Related work

In this [12] researcher described, the IDS database structure is optimized for effective data processing and storage. The Grey Wolf algorithm is used to improve feature selection and classification to increase intrusion detection accuracy. Results reveal that this integration improves intrusion detection accuracy and database administration efficiency, promising to enhance network security [13]. Emphasizes the value of a reliable database structure for the Intrusion Detection System (IDS). In order to increase the accuracy of recognizing intrusion patterns and possible threats, the study suggests a multi- stage technique for hierarchical intrusion detection. Results show how well the method enhances intrusion detection by thoroughly evaluating network activities and consequently enhancing network security.

The use of DASO Optimization Techniques was investigated by the author [14], which strongly emphasizes the value of a well-organized database for an anomaly detection system. In order to improve anomaly identification, the study uses adaptive DASO optimization approaches, demonstrating resilience to shifting network dynamics and leading to an increase in anomaly detection accuracy. By efficiently processing

data and spotting abnormalities in dynamic contexts, this method helps to increase network security. The research [15] introduces and emphasizes the importance of a robust database in supporting this approach. The study demonstrates the effectiveness of using a deep learning system powered by big data, resulting. In heightened intrusion detection accuracy by automatically learning and recognizing intricate intrusion patterns within large datasets, thereby contributing to bolstered network security. The research [16] emphasizes the importance of a well- structured database. The study demonstrates the effectiveness of a hybrid approach that amalgamates signature-based and behavior-based detection methods, improving intrusion detection accuracy by addressing a broader range of attack scenarios and anomalous behaviors, thereby enhancing overall network security.

The study [17] describes the value of an optimized database. The study demonstrates improvements in intrusion detection through the use of methodologies adapted to IoT and Edge of Things settings, improving the accuracy of spotting assaults and enhancing security for connected devices in these contexts. [18] Introduces enhancements in intrusion detection. This is done through the integration of neural networks as well as particle swarm optimization algorithms. The study combines the pattern recognition capabilities of neural networks with optimization algorithms to fine-tune detection accuracy. The results demonstrate improved intrusion detection performance. This showcases heightened accuracy in identifying intrusion patterns and enhancing overall network security.

The research [19] is using federated learning in a distributed network setting. The study proposes a novel method of intrusion detection. It emphasizes the use of distributed databases for the Intrusion Detection System's (IDS) effective data processing. The study shows how the incorporation of federated learning improves intrusion detection accuracy while resolving privacy issues and boosting network security. This is done by cooperatively training a shared model across remote devices. The research [20] presents an advanced approach for intrusion detection in large-scale data. It shows the importance of a robust database structure within the Intrusion Detection System (IDS). This also introduces a multi-stage hybrid technique that combines flow and packet-based intrusion analysis. This approach results in improved intrusion detection accuracy by comprehensively assessing network activity and patterns, enhancing overall network security in the context of big data environments.

ML techniques employed in intrusion detection systems (IDS) within the context of cyber security is being described in [21]. The study surveys various ML methods including supervised, unsupervised, and deep learning. This is outlining their suitability for different cyber threats. The results offer valuable insights into the landscape of ML-based IDS. Research is summarizing the advantages and limitations of these techniques, thereby contributing to a comprehensive understanding of how ML methods enhance intrusion detection capabilities in the realm of cyber security. A novel technique to intrusion detection is presented in the work in the [22]. The paper suggests a hybrid deep learning model. This model combines feature optimization with a well-structured database. Increased pattern recognition and higher intrusion detection accuracy are the results of this integration, which also increases network security. The work in the [23] presents a thorough method for spotting criminal activity in driverless cars. The study highlights the value of an organized database as the cornerstone of effective data management. Deep neural networks and sequential analysis stages are used in the improved multi-stage deep learning framework to identify and categorize possible threats. The results illustrate the framework's effectiveness in enhancing the security of these systems and show considerable improvements in precisely recognizing and categorizing harmful behaviors in autonomous cars.

Research in [24] explaining the dataset's crucial function in model training and assessing various algorithms for their performance in detection. The work in [25] achieves increased accuracy in intrusion detection by combining extreme learning machines with nature-inspired optimization. By producing realistic cyberattack data, a unique technique in the [26] helps to improving the ability of ML based Intrusion Detection Systems for recognizing complex threats.

IV. SELECTION OF DATABASE

A. CAP Theorem

In the earlier section, we have delivered that MySQL (SQL database) and MongoDB (NoSQL database) databases provide higher storage capacity. CAP theorem is highly useful for selecting a particular database according to the user's requirement. For this purpose, the CAP theorem is highly useful in choosing MySQL or MongoDB, according to our requirements for implementing IDS.In the term CAP, the characteristics, i.e., C is for the Consistency, A for the Availability, and P is for the Partition tolerance. In other words, the CAP

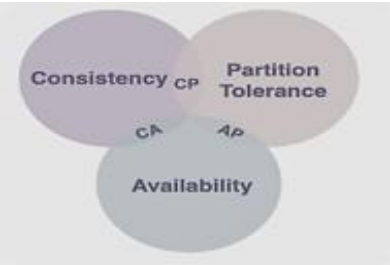

FIGURE 2. CAP theorem.

theorem presents that a distributed database system can guarantee two characteristics out of three: Consistency, Availability, and Partition Tolerance, as shown in Figure 2.

1) Consistency: A system is considered consistent if all users or clients see the same data simultaneously. Performing a read operation on consistent system must return the value of the most recent write operation. So that the read should cause all clients to return the same data which is the value of the most recent write.

2) Availability: Availability in a distributed system guarantees that the system remains operational 100% of the time. Every request gets a response regardless of the individual client.

3) Partition tolerance: This states that the system does not even fail, if the messages are delayed between clients in the system. Partition tolerance has become essential than an option in distributed systems. It is possible by replicating the records across combinations of clients and networks.

In order to select the database for a particular application, the first step is to gather application requirements based on the characteristics above, i.e., CAP. Out of these three characteristics, two of them must be satisfied. MySQL database operates on Consistency and Partition tolerance characteristics, whereas MongoDB database operates on Availability and Partition tolerance characteristics. Among others, one of our requirements is Availability and Partition tolerance. Availability ensures that our system is available to everyone at any time. Partition tolerance maintains the record and keeps the backup of the system at every time. In other words, Partition tolerance ensures no loss of data. Consequently, based on our requirements and the characteristics of the CAP theorem, we have selected the MongoDB database for the intrusion detection system.

Based on the identified common parameters in selected databases for comparison, a performance comparison has been performed, provided in Table 1. The first column of Table 1 provides the selected databases, i.e., Oracle, MySQL, Microsoft SQL Server, PostgreSQL, MongoDB, and Cassandra. Columns 2 to column 6 of Table 1 provide different parameters selected for the performance evaluation [4].

In Netspection Database, there are three core functionalities which are the core focus of development.

- Rules Parsing
- Event Engine
- Alerts

Rule parsers are used to initialize the configuration file of IDS. The first rule is dissected into different data structures, further used to detect Patterns in incoming packets. After the parsing of packets, the event engine is used to detect any malicious content. All the signatures in the rules file are first stored in data structures that reside in memory on runtime. These signatures are used to detect malicious content in incoming packets. In the end, if malware is detected, the event engine raises an alert that will be stored in the database. These stored alerts are used to visualize data on the front end. Mechanics and operational functionality of Network Intrusion Detection Systems (NIDS) is shown in Figure 3.

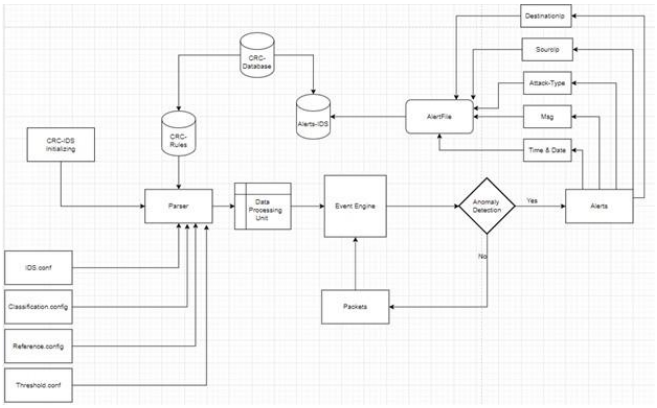

FIGURE 3. Operational functionality of NIDS.

V. METHODOLOGY

In the first step of the experiment, 10 million rules have been generated using Python script. After that, generated rules are stored in the MongoDB and script file. we considered three scenarios. Those scenarios are best, average and worst. In the best case scenario, first rule is scheduled in the text file and performed matching. After that, selected rule with the MongoDB are being matched. For both cases, the time is noted which is required for matching the rule. The required time to match from script file is 0.001s. On the other hand, 0.026s is required for matching using the MongoDB.

In the average case scenario, the rule in the middle of the script file have been selected and matching is performed. Selected rules are also matched with MongoDB. For both cases, time is noted which is required to match the rule. The time required to match from the text file is 8.49 seconds. On the other hand, 3.103 seconds is required for matching using MongoDB.

In the worst case scenario, last rule is selected in the script file and performed matching. Selected rule have been matched with the MongoDB. Time have been noted required to match the rule. The time required to match from a text file is 19.59 seconds, while 6.5 seconds is required to match using MongoDB. The performance comparison is shown in Figure 4.

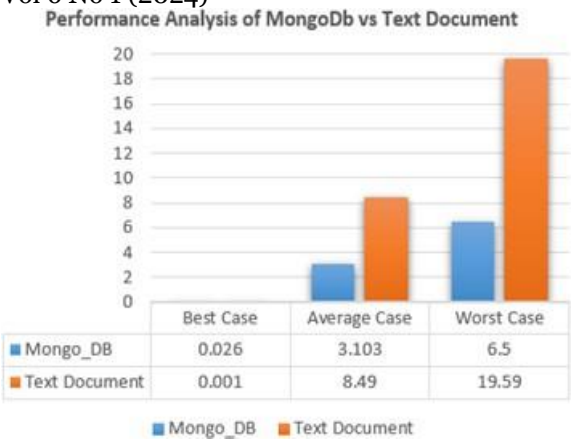

FIGURE 4. Performance comparison of MongoDB with Text File.

VI. RESULTS ANALYSIS AND DISCUSSION

We have analyzed the above-mentioned experiments and found that databases are faster than file systems to fetch and display data. Based on this experiment, we chose a database for CRC IDS. The primary purpose of the database is to store alerts and later use those alerts for better visualization on the front end. We have analyzed two databases and a file system for test purposes and found that MongoDB performed well for our system, as shown in the Figure 5.

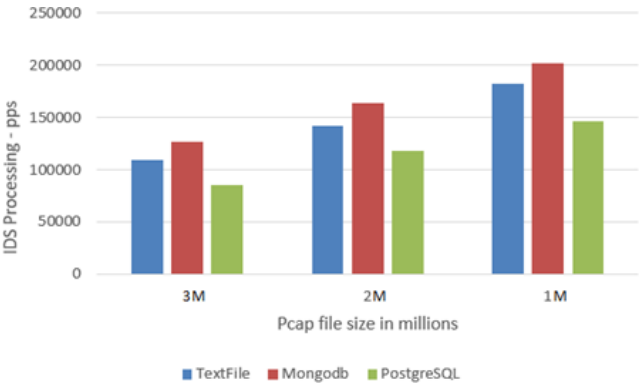

FIGURE 5. Analyzing NIDS performance across multiple databases.

Optimized queries are used to enhance the speed of data retrieval further. Buckets are used to store or fetch data rather than a single query for each alert. Calling the database for each alert is resource-extensive and slow in data retrieval. We performed multiple experiments by varying the bucket size and got the highest accuracy on a bucket size 100000. The concept of a bucket is that rather than sending each alert individually to the database, we store all alerts in a bucket first, then send bulk alerts to the database in a single call. As shown in Figure 6, 100000, bucket size has maximum efficiency; reducing or increasing bucket size has a negative impact on data retrieval speed.

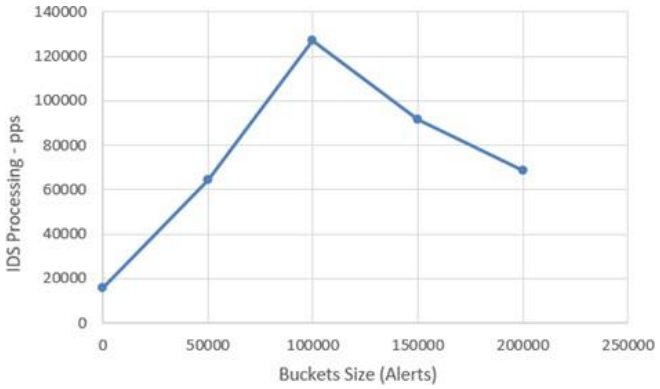

FIGURE 6. Optimizing bucket size to enhance NIDS performance.

Based on the results, it is being analyzed that the MongoDB requires less time to perform a match than the text file. Moreover, it is also analyzed that MongoDB provides a rapid response for searching a rule from many rule datasets. Consequently, the response of the search rule from the text file is three times lower than that of the MongoDB. In addition to the comparative analysis of matching speed between MongoDB and text files, we further examined the underlying reasons for MongoDB's superior performance. One key factor contributing to MongoDB's efficiency is its indexing, which allows faster data retrieval than linear search methods commonly employed with text files. Moreover, MongoDB's document-oriented structure facilitates storage as well as retrieval of complex data structures. This is making it well-suited for handling large rule datasets.

VII. CONCLUSION

The research explores about the effectiveness of Network Intrusion Detection Systems (NIDS) using a multi-stage optimized intrusion pattern. The research is focusin on the usage of databases in NIDS. Research also explores about the requirement for organized data storage. MongoDB delivers quicker rule matching and response times than text files. This is concluded according to the research of several database formats, including MongoDB and text files. This result indicates the positive effects of using an effective and well-structured database, like MongoDB for NIDS optimization. The study emphasizes about the importance of database selection and optimization for enhancing the NIDS ability to detect and address network intrusions effectively.

ACKNOWLEDGMENT

The authors of this paper want to thank everyone who has encouraged and supported people aiming to succeed in their fields. The belief of supporters in our goals has meant a lot to us and has helped us achieve what we have.

REFERENCES

[1] Y. Wang, Y. Cheng, Q. Qi, and F. Tao, “Ids-kg: An industrial dataspace-based knowledge graph construction approach for smart maintenance,” *Journal of Industrial Information Integration*, p. 100566, 2024.

[2] M. Yu, W. Zang, and P. Liu, “Database isolation and filtering against data corruption attacks,” in *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*. IEEE, 2007, pp. 97–106.

[3] K.-m. Zheng, X. Qian, and N. An, “Supervised non-linear dimensionality reduction techniques for classification in intrusion detection,” in *2010 International Conference on Artificial Intelligence and Computational Intelligence*, vol. 1. IEEE, 2010, pp. 438–442.

[4] A. Kummerow, K. Schaefer, C. Monsalve, M. Alramlawi, S. Nicolai, and P. Bretschneider, “A cyber-security testbed for the dynamic operation of transmission power systems,” in *ETG Congress 2023*. VDE, 2023, pp. 1–6.

[5] J. Pavlik and N. D. Bastian, “Cyber creative generative adversarial network for novel malicious packets,” *Proceedings in, Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications*, vol. 12529, 2023.

[6] A. Kalidindi and M. B. Arrama, “Enhancing iot security with deep stack encoder using various optimizers for botnet attack prediction,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, 2023.

[7] T. Ban, T. Takahashi, S. Ndichu, and D. Inoue, “Breaking alert fatigue: Ai-assisted siem framework for effective incident response,” *Applied Sciences*, vol. 13, no. 11, p. 6610, 2023.

[8] A. Nisioti, A. Mylonas, P. D. Yoo, and V. Katos, “From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3369–3388, 2018.

[9] A. Jannat, U. Hayat and T. Sadiq, "Exploration of Machine Learning Algorithms for Development of Intelligent Intrusion Detection Systems," *International Conference on Communication, Computing and Digital Systems (C-CODE)*, Islamabad, Pakistan, 2023, pp. 1-6, doi: 10.1109/C-CODE58145.2023.10139885.

[10] S. Myneni, K. Jha, A. Sabur, G. Agrawal, Y. Deng, A. Chowdhary, and D. Huang, “Unraveled a semi-synthetic dataset for advanced persistent threats,” *Computer Networks*, vol. 227, p. 109688, 2023.

[11] M. P. A. Saviour and D. Samiappan, “Ipfs based storage authentication and access control model with optimization enabled deep learning for intrusion detection,” *Advances in Engineering Software*, vol. 176, p. 103369, 2023.

[12] H. J. Hadi, U. Hayat, N. Musthaq, F. B. Hussain and Y. Cao, "Developing Realistic Distributed Denial of Service (DDoS) Dataset for Machine Learning-based Intrusion Detection System," *9th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, Milan, Italy, pp.1-6, 2022, doi:10.1109/IOTSMS58070.2022.10062034.

[13] M. Verkerken, L. D’hooge, D. Sudyana, Y.-D. Lin, T. Wauters, B. Vol- ckaert, and F. De Turck, “A novel multi-stage approach for hierarchical intrusion detection,” *IEEE Transactions on Network and Service Management*, vol. 20, pp. 3915-3929, 2023.

[14] S. Bhosale, A. Deshmukh, B. Deore, and P. Bhosale, “Anomaly detection through adaptive daso optimization techniques,” *Artificial Intelligence*. IntechOpen, Jan. 17, 2024. doi: 10.5772/intechopen.112421.

[15] W. Zhong, N. Yu, and C. Ai, “Applying big data based deep learning system to intrusion detection,” *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 181–195, 2020.

[16] H.-Y. Kwon, T. Kim, and M.-K. Lee, “Advanced intrusion detection combining signature-based and behavior-based detection methods,” *Electronics*, vol. 11, no. 6, p. 867, 2022.

[17] Y. K. Saheed, “Performance improvement of intrusion detection system for detecting attacks on internet of things and edge of things,” in *Artificial Intelligence for Cloud and Edge Computing*. Springer, 2022, pp. 321–339.

[18] A. Shokoohsaljooghi and H. Mirvaziri, “Performance improvement of intrusion detection system using neural networks and particle swarm optimization algorithms,” *International Journal of Information Technology*, vol. 12, pp. 849–860, 2020.

[19] Y. Tsuru, T. Kawakami, and T. Hasegawa, “Distributed network intrusion detection system using federated learning,” *IEICE Technical Report; IEICE Tech. Rep.*, vol. 122, no. 15, pp. 20–25, 2022.

[20] P, Rajesh Kanna and S S, Rajasekar and P, Santhi and G, Sathish Kumar, Mshids: Multi-Stage Hybrid Intrusion Detection System Using Flow and Packet-Based Intrusion Analysis in Big Data Environment. Preprint Available at SSRN: https://ssrn.com/abstract=4401867

[21] P. Parkar and A. Bilimoria, “A survey on cyber security ids using ml methods,” in *2021 5th International Conference on Intelligent Comput- ing and Control Systems (ICICCS)*. IEEE, 2021, pp. 352–360.

[22] A. Henry, S. Gautam, S. Khanna, K. Rabie, T. Shongwe, P. Bhattacharya, Sharma, and S. Chowdhury, “Composition of hybrid deep learning model and feature optimization for intrusion detection system,” *Sensors*, vol. 23, no. 2, p. 890, 2023.

[23] I. A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, and A. Jolfaei, “An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 469–25 478, 2021.

[24] Y. A. Al-Khassawneh, “An investigation of the intrusion detection system for the nsl-kdd dataset using machine-learning algorithms,” in 2023 IEEE International Conference on Electro Information Technology IEEE, 2023, pp. 518–523.

[25] A. Alzaqebah, I. Aljarah, and O. Al-Kadi, “A hierarchical intrusion detection system based on extreme learning machine and nature-inspired optimization,” Computers & Security, vol. 124, p. 102957, 2023.

[26] O. Sen, P. Malskorn, S. Glomb, I. Hacker, M. Henze, and A. Ulbig, “An approach to abstract multi-stage cyberattack data generation for ml-based ids in smart grids,” in 2023 IEEE Belgrade PowerTech. IEEE, 2023, pp. 01–10.