AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2) (2026)

Enhancing Network Resilience via a Cognitive AI-Driven Cybersecurity Simulation Framework and Adaptive Threat Modelling

S. Krishna Reddy[1], Thandu Bhavya[1], Perumalla Divya[1], Konda Venkatalaxmi[1], Abdel Rahman[1]

[1]Department of Computer Science and Engineering, [1]Sree Dattha Institute of Engineering and Science, Nagarjuna Sagar Road, Sheriguda, Ibrahimpatnam, Rangareddy Dist, 501510, Telangana, India.

ABSTRACT

Cybercrime continues to escalate as a global security concern, encompassing various malicious activities such as phishing, malware propagation, botnet operations, and Denial of Service (DoS) attacks. Among these threats, DoS attacks are especially disruptive, as they overwhelm targeted servers with excessive traffic, resulting in service unavailability for legitimate users. Conventional defense mechanisms, including firewalls and antivirus solutions, offer limited protection and often fail to effectively mitigate large-scale traffic flooding attacks. To overcome these challenges, this study proposes a proactive DoS attack simulation and mitigation framework based on real-time network traffic monitoring and intelligent filtering. The proposed system integrates a network packet monitoring mechanism with a Big Data processing framework utilizing Hadoop MapReduce to analyze incoming traffic efficiently. Each request is evaluated by comparing its size with the server's processing capacity. Requests that fall within the defined threshold are forwarded to the server, while unusually large or suspicious requests are classified as potential DoS attempts and discarded before reaching the system, thereby safeguarding server performance. The framework consists of three major components: the Server Module, which continuously listens for authenticated upload and download requests; the Network Monitor Module, which employs MapReduce to process and analyze high-volume packet data in parallel; and the User Simulation Module, which generates both normal and malicious traffic to evaluate system robustness. Experimental results demonstrate that legitimate requests are processed successfully, whereas attack-like traffic is effectively detected and blocked. The use of Hadoop MapReduce ensures scalability and high-performance traffic analysis, while a monitoring dashboard provides visual insights into normal and malicious packet activity. The proposed approach effectively mitigates DoS attacks, prevents server disruption, and ensures service availability for legitimate users.

Key words: Network Traffic Visualization, Adaptive Threat Modelling, Cyber Range Simulation, Cognitive Cybersecurity Systems, Intelligent Cyber Defense Mechanisms

1. INTRODUCTION

Cybersecurity is a constantly evolving field that faces increasingly sophisticated challenges and ever-expanding threats. Information security and protection against cyberattacks have become critical priorities for governments, businesses, and users in a digitally interconnected world. The increasing complexity of cyber threats has led to the need for advanced and practical solutions that can anticipate, detect, and mitigate risks in real time. For this reason, artificial intelligence (AI) has established itself as a fundamental tool to strengthen cybersecurity. Despite technological advances, the continued adaptation of security strategies in response to emerging threats remains a crucial challenge, highlighting the importance of bridging theory and practice. The work presents an approach in the field of cybersecurity, combining AI with Amazon Web Services (AWS) cloud resources to create a training environment. As shown in fig 1 this collaboration represents a significant milestone and a step forward in the ongoing effort to strengthen cybersecurity through innovation and the practical application of new technologies.

Modern network environments face sophisticated cyber threats, making traditional, static security solutions insufficient, especially for training purposes where hands-on experience is limited. The primary problem is the lack of an interactive, controllable, and visually intuitive platform that can simulate network attacks and demonstrate real-time monitoring and detection processes. Without such a simulator, security professionals and trainees struggle to bridge the gap between theoretical knowledge and practical understanding of network anomaly detection, hindering their ability to effectively recognize and respond to dynamic security incidents.

Fig 1. Cyber security threat monitoring management

The motivation stems from the critical need for effective and practical cybersecurity training that reflects real-world network dynamics. By integrating a network simulation with a data processing framework like MapReduce, the project aims to demystify how large volumes of network traffic are analyzed for security breaches. This allows trainees to actively participate in generating "attack" traffic (e.g., large file transfers exceeding a threshold) and immediately witness its impact on the system's "attack packets" counter and the dashboard. This hands-on, cause-and-effect learning accelerates the development of crucial analytical and monitoring skills required in a security operations center (SOC).

2. LITERATURE SURVEY

2.1 Cyber Range, Simulation, and Wargaming Approaches

Chowdhury and Gkioulos [1] explored simulation systems positioned between cyber ranges (CRs) and tabletop exercises (TTXs), categorizing them based on their relevance to management training. Their work emphasized strategic planning tools, while Holik et al. [2] focused on technically realistic simulations for operator training. Unlike these approaches, recent methodologies prioritize decision-making at the management level, enabling scalable training without requiring deep technical infrastructure.

Johnson [6] identified multiple levels of wargaming, ranging from tactical to strategic, highlighting varying levels of abstraction and detail. Perla [7] distinguished wargames from traditional exercises by their outcome-driven progression, where scenarios evolve based on participant actions. Švábenský [8] further enhanced this concept by integrating cyber ranges with serious games to improve engagement and learning outcomes.

2.2 Cybersecurity Training, Roles, and Evaluation Metrics

Ukwandu et al. [3] proposed a taxonomy identifying seven distinct participant roles in cyber exercises, reflecting the complexity of collaborative environments. Ostby et al. [4] emphasized the importance of auxiliary roles in enhancing exercise effectiveness, while Granåsen and Andersson [5] introduced performance metrics to evaluate training outcomes. These studies collectively highlight the need for structured evaluation and role-based design in cybersecurity training systems.

2.3 Organizational and Strategic Frameworks

civilian organizational hierarchy [9] described hierarchical structures in both civilian and military contexts, ranging from operational to strategic levels. This classification underscores the importance of

aligning cybersecurity training approaches with organizational decision-making layers, particularly at higher strategic levels.

2.4 Machine Learning-Based Intrusion Detection Systems

Panda et al. [10] implemented a Naïve Bayes classifier using the KDD'99 dataset, achieving high detection accuracy across multiple attack categories with a low false positive rate. Amor et al. [11] developed a Bayesian network-based framework for anomaly detection, demonstrating strong performance for certain attack types but lower accuracy for rare attack classes. Kokila et al. [12] utilized Support Vector Machines (SVM) for detecting DDoS attacks in software-defined networks, achieving high accuracy but facing computational limitations. Amiri et al. [13] improved SVM efficiency by reducing feature dimensionality, achieving high classification accuracy across multiple attack types.

Hu et al. [14] proposed a robust SVM model that enhances generalization by smoothing the decision boundary, demonstrating improved accuracy with controlled false positive rates. Vuong et al. [15] applied decision tree models to detect cyberattacks in robotic systems, effectively identifying different attack behaviors based on their impact.

3. PROPOSED SYSTEM

The designed system follows a structured pipeline that combines simulation, distributed analytics, anomaly identification, and real-time visualization to deliver an interactive cybersecurity training environment. As shown in fig 2 in this framework, an AI-enabled training simulator connects a network traffic generation module with a distributed processing mechanism inspired by MapReduce to support continuous monitoring. Trainees initiate simulated file transmissions through a designated communication port (e.g., 3333), producing network traffic that is separately observed and processed by a monitoring unit operating on another port (e.g., 4444). The analytical component applies a simplified rule-based logic, where data packets are evaluated primarily based on attributes such as size—to determine whether they represent legitimate activity or potential threats. This classification process dynamically updates system metrics, enabling immediate feedback. The interpreted outcomes are then presented through an interactive visualization interface powered by JFreeChart, allowing users to observe system behavior in real time and understand how distributed computation combined with basic intelligent rules can effectively identify and react to suspicious network events.

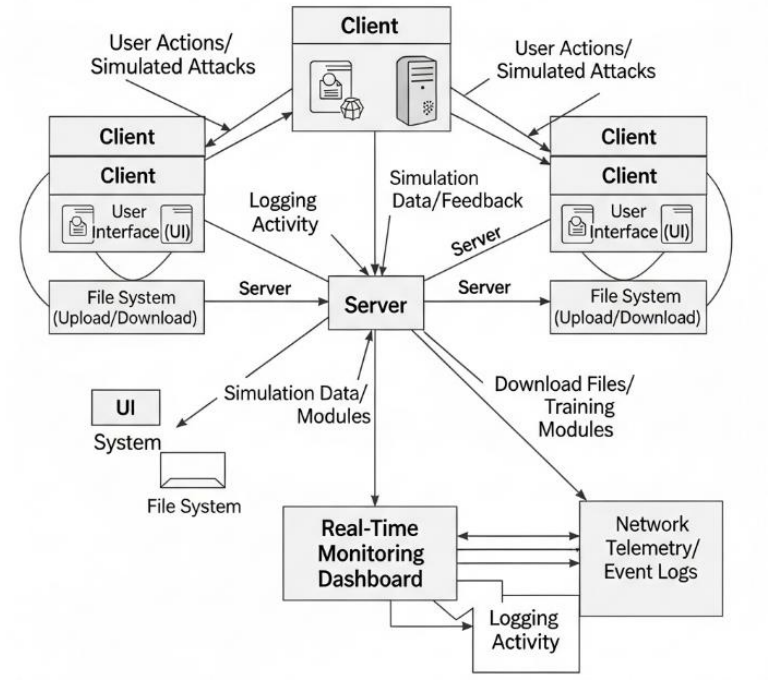

Fig. 2. Proposed system architecture of cyber security training simulator

The process begins with the Main and Network classes providing a graphical user interface where a user configures a network with simulated users. The user then initiates a file transfer (upload or download) via the Network class, selecting a file size and a simulated user ID. This action triggers two key concurrent processes: the file transfer itself handled by the Server (Port 3333) and the generation of network traffic data representing the file size, which is sent to the monitoring component. A dedicated component, the Network Monitor (Port 4444), runs as a separate server listening for traffic flow information generated by the simulation. Upon receiving this data (the simulated file size), the monitor component initiates the core analysis by calling the simulated MapReduce job. This separates the function of secure data storage/transfer from the function of security monitoring and analysis, mimicking a decoupled enterprise environment. The received traffic data (file size) is passed to the Hadoop runner class, which simulates the execution of a distributed job. The core security logic resides in the MapReduce class, specifically its map function. Here, the raw size data is converted to megabytes, and the AI-driven logic (currently a simple rule: size > 50 MB) is applied to determine the packet status. If the size exceeds the threshold, the traffic is flagged as "abnormal" (attack), otherwise, it is flagged as "normal." This step updates static counters (Network Monitor.normal and Network Monitor.attack) in real-time.

As the MapReduce process classifies packets and updates the counters, the monitoring results are immediately ready for display. The user can click a button on the GUI to launch the Chart class. This class uses the JFreeChart library to dynamically pull the updated normal and attack packet counts from the Network Monitor and render them into a Comparison Bar Graph. This dashboard provides an instantaneous visual feedback loop, allowing the trainee to observe the direct impact of their simulated actions (e.g., uploading a large "attack" file) on the network's security status. Separately, the Server and Process Thread handle the actual file upload/download requests. Importantly, the system can demonstrate a security response: if a simulated attack is detected (or explicitly commanded by the user), the Process Thread logs a message indicating that the file is being ignored due to "Malicious Content detected," completing the training cycle of detection and response within the simulated environment.

Map Reduce Framework

The Simulated MapReduce Framework to execute the real-time anomaly detection logic. This component is crucial for demonstrating how a distributed processing system handles continuous network monitoring. The Hadoop class initiates the job, passing the single piece of traffic data (the file size) to the MapReduce class. The core work is done entirely within the Mapper function, which parses the file size, normalizes it into megabytes, and applies a simple rule-based model (size) to classify the traffic as normal or attack. Critically, the Mapper bypasses the traditional Reducer and directly updates the static counters in the NetworkMonitor class, enabling immediate, real-time feedback and visualization on the JFreeChart dashboard.

The MapReduce component is simulated to demonstrate how a distributed framework would analyze network traffic for anomalies. Its workflow is executed entirely within the single MapReduce class, triggered by the Hadoop class, using the single file size value as input. The external Hadoop.run(String path) method first initializes the MapReduce job configuration (JobConf). The input path is set to point to the file containing the traffic data (simulated to be just the file size). In a real Hadoop cluster, the system would split the input file into chunks, but here, the input is functionally just a single line representing the file size, ensuring the entire "job" focuses on this one data point

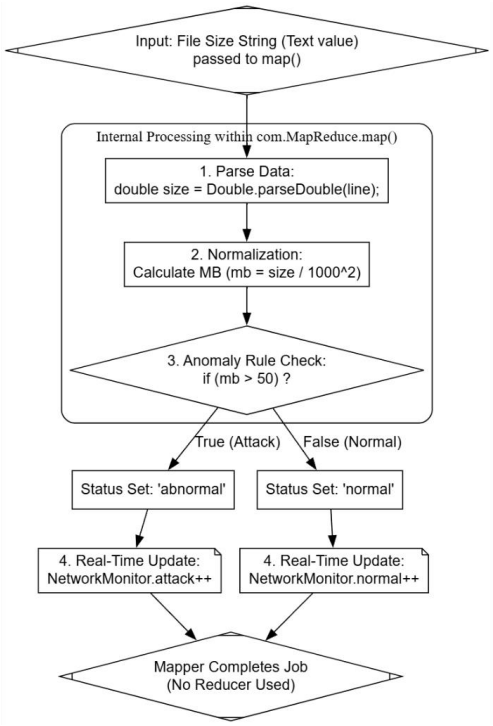

Fig 3. Internal workflow of Map Reduce Framework

As shown in fig 3 the MapReduce framework instantiates the MapReduce class, which acts as the Mapper. The framework then calls the primary processing method, map (LongWritable key, Text value, OutputCollector<Text,Text> output, Reporter reporter), passing the single line of input (the file size string) as the value parameter. The Mapper immediately extracts this string into a local variable named line. The Mapper's internal logic begins the transformation of the raw data. The line (the file size string) is first converted into a double-precision number: double size = Double.parseDouble(line);. This raw size is assumed to be in bytes. It then performs normalization to convert the size into megabytes (MB) via two steps. This is the core decision-making stage. The Mapper applies the simple, rule-based detection logic: if(mb > 50). This condition determines the status:

- If the file size exceeds MB, the traffic is deemed "abnormal," signifying a potential attack.
- If the file size is MB or less, the traffic is deemed "normal."

Based on the classification, the Mapper directly updates the static variables within the NetworkMonitor class, bypassing the traditional MapReduce Reducer phase to ensure immediate, real-time feedback to the dashboard If attack, it executes: NetworkMonitor.attack = NetworkMonitor.attack + 1;

If normal, it executes: NetworkMonitor.normal = NetworkMonitor.normal + 1; The overall status is also set: NetworkMonitor.status = status;. This is the crucial step that simulates real-time monitoring and makes the detection visible to the GUI. After processing the single input line, the map method returns, and since there is no Reducer specified or needed for this simple classification and direct static update, the MapReduce job completes. The updated NetworkMonitor counters are immediately available to the Chart class for dashboard visualization

4. RESULT DESCRIPTION

Fig 4 depicts a cyber security simulation interface designed for monitoring network activity. Multiple nodes, representing users, are displayed as green and yellow circles labeled with IDs (e.g., U1, U2, U12), indicating active participants in the network. At the top, buttons for “Server” and “Network Monitor” suggest interactive controls for viewing server status and monitoring traffic. A file upload dialog is open, allowing a user to select a file (“video.avi”) for upload, highlighting the simulator’s functionality for file transfer and network monitoring. At the bottom, controls for selecting user ID, file

name, and buttons for uploading, downloading, and exiting the simulation are visible, providing real-time interaction and operational testing of the cyber security environment. The visual layout demonstrates a dynamic, interactive system for testing network security scenarios.

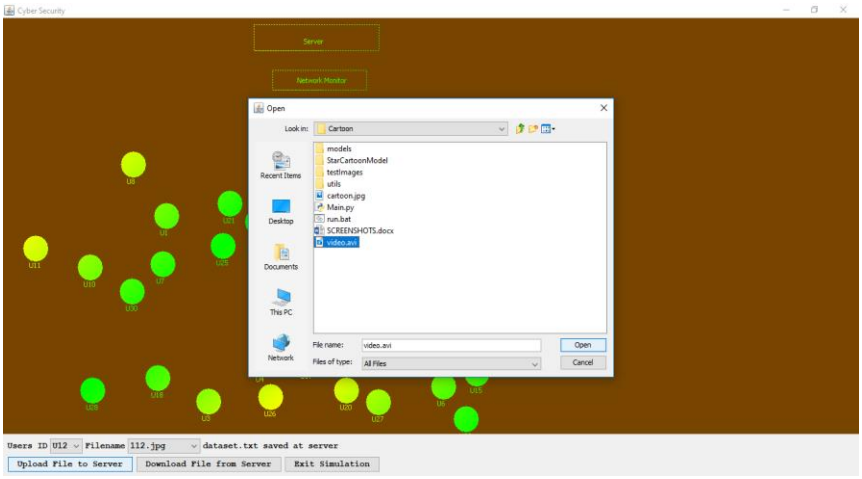

Fig. 4. File Uploading interface for threat monitoring

Fig 5 shows the same cybersecurity simulation screen as before, focused on user nodes represented by green and yellow circles, labeled with user IDs (such as U12, U18, U9, etc.). The top contains buttons for "Server" and "Network Monitor." At the bottom, a text log shows the message "video.avi size out of limit and discarded from storage," alongside a user selection of "ID U12" and a file name "112.jpg." The buttons to upload files to the server, download files from the server, and exit the simulation are visible but inactive in this screenshot. Unlike the previous images, no popup message box is displayed here, and the desktop taskbar is also partially visible

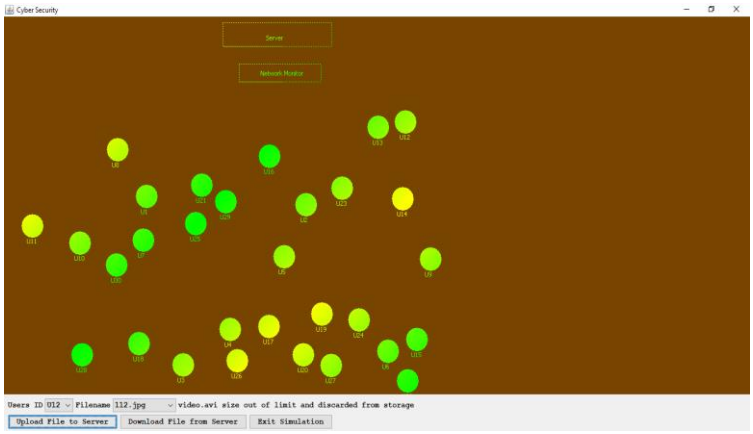

Fig. 5. Network User Nodes simulation and Storage Log

Fig 6 displays a network simulation likely related to Cyber Security, as indicated by the window title. It shows a Server connected to a Network Monitor, with several dispersed nodes labeled 'U' (U01, U02, ..., U14, etc.) representing Users. These user nodes are color-coded in shades of green and yellow, which could represent various states like security level, activity, or data transfer status within the network. At the bottom, a status bar indicates "User ID U12 Filename 112.jpg : User : U12 requesting file 112.jpg", confirming a file request is in progress. Controls for "Upload File to Server", "Download File from Server", and "Exit Simulation" are also visible

AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2) (2026)

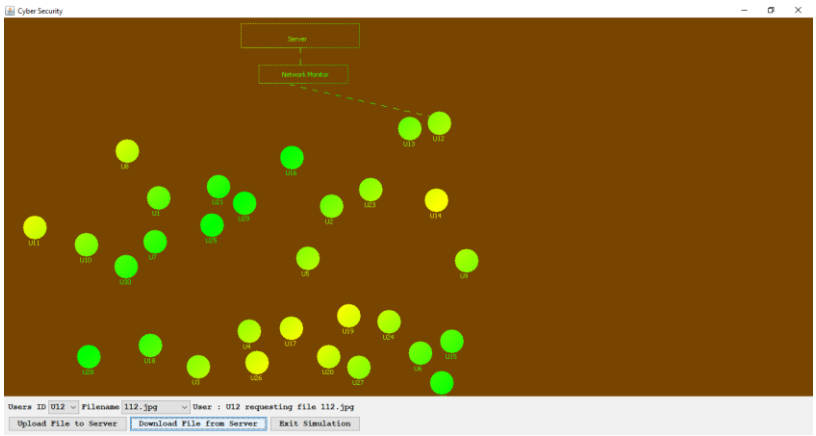

Fig. 6. Network Environment illustrating a User File Request

Fig 7 displays network packet monitoring map reduce server application, centered on a "Normal & Attack Packets Comparison Graph." The bar chart visually represents network security data, showing that the volume of "Total Normal Packets" (blue bar, count) is currently higher than the volume of "Total Attack Packets" (green bar, count). The console output displayed above the graph indicates the "Network Monitor MapReduce Server Started," logging connections from 127.0.0.1 and detailing file activity, specifically noting a file size "under limit" for storage and another file, video.avi, whose size is "out of limit," demonstrating the system's role in both traffic analysis and file security monitoring.

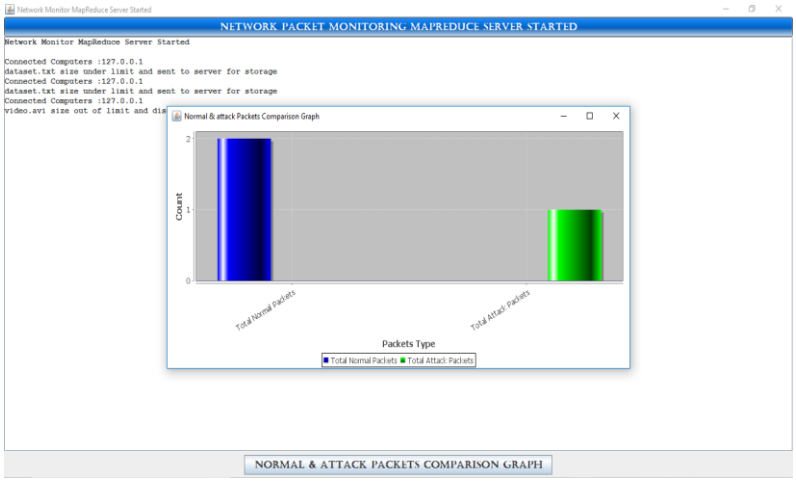

Fig. 7. Comparison Graph of Normal and Attack Packets Detected by the MapReduce Server.

Fig 8 presents the console output specifically functioning as the "Cyber Security Storage Server." This window serves as a transaction log, documenting the server's operational status and recent activity. The log confirms that the "Cyber Security Server Started" and records multiple successful "Connected Computers :127.0.0.1" events. Crucially, it tracks file management operations, repeatedly noting that "dataset.txt saved at server" and confirming the successful delivery of a file with the entry "112.jpg File sent to user 012," which aligns with the observed download actions from the network simulation. In summary, the console validates the server's connectivity, data storage, and the successful delivery of the requested file to user U12

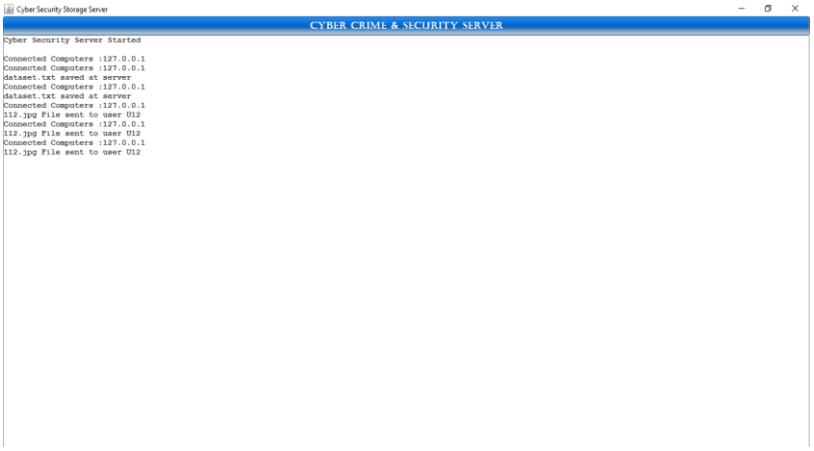

Fig. 8. Console Log of Cyber Crime & Security Server Activity and File Transactions

5. CONCLUSION

The AI-driven Cyber Security Simulator successfully integrates network monitoring, MapReduce-based traffic analysis, and AI-based anomaly detection into a single platform. The project provides a realistic simulation of multiple network users performing file uploads and downloads while continuously monitoring packet-level traffic for malicious activity. Real-time feedback, secure file handling, and visual representation of network behavior enhance the understanding of cyber security dynamics. The system effectively detects abnormal traffic, prevents malicious file storage, and demonstrates the practical utility of combining AI and MapReduce for network security. Overall, the project achieves its goal of creating an interactive, automated, and intelligent cyber security training and monitoring environment.

REFERENCES

[1]. Chowdhury, N.; Gkioulos, V. Cyber security training for critical infrastructure protection: A literature review. *Comput. Sci. Rev.* **2021**, *40*, 100361.

[2]. Holik, F.; Yayilgan, S.Y.; Olsborg, G.B. Emulation of Digital Substations Communication for Cyber Security Awareness. *Electronics* **2024**, *13*, 2318.

[3]. Ukwandu, E.; Farah, M.A.B.; Hindy, H.; Brosset, D.; Kavallieros, D.; Atkinson, R.; Tachtatzis, C.; Bures, M.; Andonovic, I.; Bellekens, X. A review of cyber-ranges and test-beds: Current and future trends. *Sensors* **2020**, *20*, 7148.

[4]. Ostby, G.; Lovell, K.N.; Katt, B. EXCON Teams in Cyber Security Training. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 5–7 December 2019; IEEE: New York, NY, USA, 2019; pp. 14–19.

[5]. Granåsen, M.; Andersson, D. Measuring team effectiveness in cyber-defense exercises: A cross-disciplinary case study. *Cogn. Technol. Work* **2016**, *18*, 121–143.

[6]. Johnson, J. The "Four Levels" of Wargaming: A New Scope on the Hobby. 2014. Available online: **https://www.beastsofwar.com/featured/levels-wargames-exploring-scopes-hobby/** (accessed on 30 August 2024).

[7]. Perla, P. *The Art of Wargaming: A Guide for Professionals and Hobbyists*; Naval Institute Press: Annapolis, MD, USA, 2012.

[8]. Švábenský, V.; Cermak, M.; Vykopal, J.; Laštovička, M. Enhancing cybersecurity skills by creating serious games. In Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE, Larnaca, Cyprus, 2–4 July 2018; pp. 194–199.

[9]. Mihm, J.; Loch, C.H.; Wilkinson, D.; Huberman, B.A. Hierarchical Structure and Search in Complex Organizations. *Manag. Sci.* **2010**, *56*, 831–848.

[10]. Panda, M.; Patra, M.R. Network intrusion detection using naive bayes. *Int. J. Comput. Sci. Netw. Secur.* **2007**, *7*, 258–263.

[11]. Amor, N.B.; Benferhat, S.; Elouedi, Z. Naive bayes vs decision trees in intrusion detection systems. In Proceedings of the 2024 ACM Symposium on Applied Computing, Nicosia, Cyprus, 14–17 March 2024; pp. 420–424.

[12]. Kokila, R.; Selvi, S.T.; Govindarajan, K. DDoS detection and analysis in SDN-based environment using support vector machine classifier. In Proceedings of the 2014 Sixth International Conference on Advanced Computing (ICoAC), Chennai, India, 17–19 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 205–210. [Google Scholar]

[13]. Amiri, F.; Yousefi, M.R.; Lucas, C.; Shakery, A.; Yazdani, N. Mutual information-based feature selection for intrusion detection systems. J. Netw. Comput. Appl. 2011, 34, 1184–1199. [Google Scholar] [CrossRef]

[14]. Hu, W.; Liao, Y.; Vemuri, V.R. Robust Support Vector Machines for Anomaly Detection in Computer Security. In Proceedings of the ICMLA, Los Angeles, CA, USA, 23–24 June 2003; pp. 168–174. [Google Scholar]

[15]. Vuong, T.P.; Loukas, G.; Gan, D.; Bezemskij, A. Decision tree-based detection of denial of service and command injection attacks on robotic vehicles. In Proceedings of the 2015 IEEE International Workshop on Information Forensics and Security (WIFS), Rome, Italy, 16–19 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–6.