Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 2, Issue 5, 2024

AI-BASED MODEL FOR CYBERSECURITY: IDENTIFYING THREATS

Shujaat Ali Rathore[*1], Muhammad Hammad u Salam[2], Dr. Mohd Yaqoob Wani[3], Mehmood Ashraf[4], Muhammad Irfan[5]

*[*1]Department of Computer Science & Information Technology, University of Kotli, Azad Jammu and Kashmir. Corresponding Author*
[2]Department of Computer Science & Information Technology, University of Kotli, Azad Jammu and Kashmir.
[3]Dean, Faculty of Computer Sciences, Ibadat International University, Islamabad,
[4]Department of Communication and Cyber Security, Bahauddin Zakariya University, Multan, 60000, Pakistan,
[5]Department of Computer Science, NCBA&E, Sub-Campus Multan, 60000, Pakistan

shujaat.ali@uokajk.edu.pk

DOI: https://doi.org/10.5281/zenodo.17165509

Keywords
Cybersecurity, Intrusion detection systems, artificial intelligence, Machine learning, Binary Grasshopper Optimized twin support vector machine

Article History
Received: 15 October2024
Accepted: 13 December 2024
Published: 28 December 2024

Corresponding Author: *
Shujaat Ali Rathore

Abstract
The challenge of maintaining cyber-security has been intensifying due to the rapid growth in computer interconnectivity and the increasing variety of computer-based applications in recent years. To counter the rising wave of cyber threats, systems require strong and adaptive defenses. One effective solution is the use of Intrusion Detection Systems (IDS), which help identify anomalies and potential risks within computer networks. In this research, an advanced data-driven IDS is developed using Artificial Intelligence, with a particular focus on Machine Learning techniques. A new security framework, the Binary Grasshopper Optimized Twin Support Vector Machine (BGOTSVM), is introduced. This model prioritizes and ranks security features based on their importance before constructing the IDS with only the most relevant features. By reducing feature dimensions, the method not only enhances prediction accuracy for unseen data but also decreases computational cost. To evaluate its efficiency, experiments are carried out using four well-known ML methods–Decision Tree, Random Decision Forest, Random Tree, and Artificial Neural Network–then compared with existing approaches. The experimental results validate that the proposed technique can function as a reliable learning-based model for network intrusion detection and show strong potential for real-world implementation.

INTRODUCTION

In recent years, the use of computer and network technologies has expanded widely, involving private, public, and commercial data. This growth has made cyber security increasingly vital to prevent system intrusions. Earlier, setting up a firewall security policy was often insufficient, as newer forms of attacks exploited operating system weaknesses, message exchange parameters, and other vulnerabilities. However, with the application of Intrusion Detection Systems (IDS), it is possible not only to detect these issues but also to block unauthorized access [1].

Cyber security has advanced significantly as a response to the rising number of cyber threats, aiming to counter cybercrime. It represents a combination of technologies, skilled professionals, and procedures designed to establish protective mechanisms that safeguard cyberspace from malicious actors. Cyber security methods can generally be classified into two

categories: conventional approaches and automated approaches. Traditional methods often face limitations such as poorly trained personnel, inadequate system design, and limited access to reliable data, all of which contribute to the rise in cybercrimes [2].

The emergence of Artificial Intelligence (AI) has enhanced learning-based techniques for detecting cyberattacks, with numerous studies showing promising results. Nonetheless, securing IT infrastructures against evolving attacks and suspicious network activity remains highly challenging. Because of repeated network breaches and the severe damage they cause, significant attention has been placed on developing robust defense mechanisms and reliable security solutions [3].

At the same time, cybercriminals have continued to expand their malicious operations, exploiting new vulnerabilities and bypassing security protections to infiltrate secure communication networks. Such activities cause a range of damages, from service disruptions to theft of confidential, sensitive, or financial data. The rapid increase in cyber-attacks—particularly AI-assisted hacking—reflects the overall expansion of the digital ecosystem. These attacks have recently emerged as a new threat to already overburdened cybersecurity practices, many of which depend on costly human intervention [4].

The structure of this study is as follows: Section II presents related work, Section III explains the proposed methodology, Section IV discusses experimental results, and Section V concludes with recommendations for future work.

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

The current system for the project "Cyber Security Threat Detection Model using Artificial Intelligence Technology" applies Machine Learning methods, with a specific focus on the Binary Grasshopper Optimized Twin Support Vector Machine (BGOTSVM) approach. Its purpose is to tackle the growing challenges of cybersecurity by identifying anomalies and malicious activities within computer networks. The workflow begins with data collection and preprocessing, followed by ranking security-related features based on their importance. Using these key features, the Intrusion Detection System (IDS) is built, which reduces both dimensionality and computational overhead. Performance comparisons are made with conventional ML models such as Decision Tree, Random Decision Forest, Random Tree, and Artificial Neural Network. The evaluation results highlight that the system achieves better accuracy in intrusion detection and shows promise for real-world security implementations.

LIMITATIONS

Limited Scalability: The system may encounter difficulties when applied to large-scale and complex network infrastructures, reducing its effectiveness for organizations with broad and diverse networks.

High Dependency on Feature Selection: Since system performance depends greatly on the feature selection process, non-optimal selection can result in false positives or false negatives.

Lack of Real-Time Detection: Although effective, the system may not provide full real-time monitoring or instant alerts, which are essential for timely responses to cyber threats.

Low Interpretability: Advanced models like BGOTSVM are often difficult to interpret, making it challenging for cybersecurity experts to fully understand or explain the system's decision-making process.

Insufficient Continuous Adaptation: Cyber threats evolve constantly, but the system lacks an inbuilt mechanism for automatic updates or adaptive learning, leaving potential gaps against newly emerging attacks.

2.2 PROPOSED SYSTEM:

The proposed framework for the project *"Cyber Security Threat Detection Model using Artificial Intelligence Technology"* is designed to overcome the shortcomings identified in the existing system. To improve scalability, it integrates distributed computing along with cloud-based solutions, making it adaptable to extensive and complex network environments. The system automates the feature selection process through advanced feature engineering methods, minimizing dependency on manual selection. Real-time detection mechanisms are introduced to enable rapid identification and response to cyber threats. For greater transparency, explainable AI techniques are applied so that system decisions are more

comprehensible to cybersecurity specialists. Furthermore, the system emphasizes adaptability by incorporating continuous learning mechanisms and threat intelligence feeds, ensuring it remains updated and resilient against newly emerging cyber threats.

2.3 ADVANTAGES

Improved Scalability: The system is capable of scaling effectively to manage extensive and complex network setups, making it well-suited for organizations with large and diverse infrastructures.

Automated Feature Engineering: Through automation of feature selection, the system minimizes human error and guarantees consistent identification of the most critical security features.

Real-Time Threat Response: With built-in real-time monitoring, the system can instantly detect threats and allow organizations to react quickly, thereby reducing potential risks and damages.

Better Interpretability: The integration of explainable AI methods enhances system transparency, enabling cybersecurity experts to clearly understand and rely on its decision-making process.

Ongoing Adaptability: Emphasizing continuous learning and adaptability, the system ensures regular updates to counter new and evolving cyber threats, providing stronger and more resilient security.

SYSTEM IMPLEMENTATION

3.1 Modules

Data Collection and Preprocessing

Feature Engineering and Selection

Model Development

Real-Time Monitoring and Alerting

Explainability and Adaptation

3.2 Data Collection and Preprocessing: This module is responsible for gathering network traffic data, system logs, and activity parameters. The collected data is then cleaned and standardized to maintain quality and consistency before being passed on for further analysis.

3.3 Feature Engineering and Selection: Advanced feature engineering techniques are applied to extract and refine critical security-related attributes. Automated feature selection ensures that only the most relevant features are prioritized, minimizing redundancy and improving detection accuracy.

3.4 Model Development: The Intrusion Detection System (IDS) is built using the Binary Grasshopper Optimized Twin Support Vector Machine (BGOTSVM) model. The algorithm is optimized to achieve higher accuracy, efficiency, and reliability in detecting cyber threats.

3.5 Real-Time Monitoring and Alerting: This module integrates continuous monitoring of network activity to promptly identify malicious or suspicious behavior. On detection, instant alerts are generated to enable rapid responses and minimize potential risks.

3.6 Explainability and Adaptation: To ensure transparency, explainable AI methods are incorporated so that the model's decisions can be clearly understood by cybersecurity professionals. In addition, adaptive mechanisms are included for continuous learning, supported by threat intelligence feeds and regular updates, allowing the system to remain resilient against evolving attacks.

Hardware Requirements

Processor: Intel i3, 2.2 GHz

RAM: 4 GB

Storage: 256 GB Hard Disk

SOFTWARE REQUIREMENTS

Operating System (OS): Windows 10 or 11

Programming Language: Python

Tool: Visual Studio Code (VS Code)

SOFTWARE ENVIRONMENT

4.1 Python

Python is a high-level, structured, open-source programming language widely used for diverse programming applications. It functions as an interpreted language, automatically compiling the code into bytecode before execution. Being dynamically typed, it offers flexibility and also supports object-oriented programming features without requiring them explicitly.

Python has been adopted in various large-scale applications due to its reliability and versatility. For instance, NASA utilizes Python for several of its software systems and has standardized it within its Integrated Planning System. Similarly, major technology companies employ Python extensively—Google leverages it in the development of its Web Crawler and Search Engine components, while

Yahoo! applies it in managing its online discussion groups.

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

SYSTEM TEST

The primary objective of testing is to uncover errors and weaknesses in a system. It involves applying various methods to identify faults in components, subassemblies, complete assemblies, and the final product. Testing ensures that the developed software meets its defined requirements, aligns with user expectations, and operates reliably without critical failures. Different testing approaches are used, each serving a specific purpose in validating the system.

TYPES OF TESTS

5.1 Unit Testing

Unit testing verifies the correctness of internal program logic by ensuring that given inputs produce valid outputs. It validates all decision branches and internal code flows. Conducted once individual units are complete, this form of testing is structural and invasive, relying on the system's internal construction. Unit testing focuses on specific modules or processes, ensuring that each unique execution path works as defined in the specifications with clear inputs and expected outputs.

5.2 Integration Testing

Integration testing evaluates whether combined software modules interact correctly as a single system. While individual components may function properly in isolation (as proven by unit tests), integration testing exposes issues that arise when modules are combined. It validates event-driven outcomes, such as screen transitions and data exchanges, ensuring consistency across the integrated system.

5.3 Functional Testing

Functional tests systematically confirm that all required features and business processes work as documented in requirements, technical specifications, and user manuals. They validate correct handling of both valid and invalid inputs, exercise all defined functions, verify outputs, and test interfacing systems or procedures. The focus is on ensuring that business workflows, data handling, and process sequences operate as expected. Additional test cases may be introduced before completion to improve overall coverage and effectiveness.

5.4 System Testing

System testing validates the behavior of the entire integrated software solution against its requirements. It ensures the system configuration delivers consistent and predictable results. This level of testing emphasizes process flows, integration points, and configuration-based validations, such as system integration checks.

5.5 White Box Testing

White Box Testing requires knowledge of the system's internal structure, logic, and language. It is applied to test areas inaccessible from a black-box perspective, allowing testers to validate internal processes, execution paths, and logic flows.

5.6 Black Box Testing

Black Box Testing treats the software as a closed system where internal workings are unknown to the tester. Inputs are provided, and outputs are observed without considering how the system processes them. Test cases are derived from requirement documents or specifications, ensuring functionality matches defined expectations.

Test Strategy and Approach

Field testing will be performed manually, and functional test cases will be documented in detail.

6.1 Test Objectives

Ensure all field entries operate correctly.

Verify that all pages are accessible via their designated links.

Confirm that input screens, messages, and responses load without delays.

6.2 Features to be Tested

Validate input formats.

Prevent duplicate entries.

Confirm that all links direct users to the correct pages.

6.3 Integration Testing

Incremental integration testing verifies that multiple software modules function seamlessly on a shared platform. Its main purpose is to identify and resolve interface-related defects between modules or software applications, ensuring smooth interaction at both the component and enterprise levels.

Test Results:

All test cases executed successfully, and no defects were identified.

6.4 Acceptance Testing

User Acceptance Testing (UAT) is essential in confirming that the developed system satisfies functional requirements and aligns with user expectations. It involves active participation from end users to validate real-world performance.

Test Results:

All acceptance test cases passed successfully, with no defects encountered.

CONCLUSION

IT specialists, e-commerce professionals, and application developers remain highly concerned about the practicality and reliability of machine learning–based intrusion detection models for strengthening security. Cybersecurity datasets generally contain multiple categories of attacks with diverse features, and as a result, certain classifiers may struggle to achieve high accuracy and reliable prediction rates across all attack types. This study has examined the performance of the BGOTSVM model, evaluating its effectiveness through metrics such as recall, F1-score, and overall accuracy. Looking ahead, the objective is to expand cybersecurity datasets and design a robust data-driven intrusion detection system, ultimately providing automated security solutions to assist the wider community of cybersecurity experts.

REFERENCES

P. Sornsuwit,and S. Jaiyen, "A new hybrid machine learning for cybersecurity threat detection based on adaptive boosting,"Applied Artificial Intelligence, 33(5), pp.462-482, 2019.

K.Shaukat, S. Luo, S.Chen, and D. Liu, "Cyber threat detection using machine learning techniques: A performance evaluation perspective,"inIEEE international conference on cyber warfare and security. IEEE, October2020,pp. 1-6.

Q.H. Vu, D.Ruta, and L.Cen, "Gradient boosting decision trees for cyber security threats detection based on network events logs," in 2019IEEE International Conference on Big Data (Big Data). IEEE, December2019, pp. 5921-5928.

J. Lee, J. Kim, I.Kim, and K. Han,"Cyber threat detection based on artificial neural networks using event profiles,"IEEE Access, vol. 7, pp.165607-165626, 2019.

J.H. Li, "Cyber security meets artificial intelligence: a survey;" Frontiers of Information Technology & Electronic Engineering, vol. 19,no.12, pp.1462-1474, 2018.

N.Rawindaran, A.Jayal, E.Prakash, and C.Hewage, "Cost benefits of using machine learning features in NIDS for cyber security in UK small medium enterprises (SME)," Future Internet, vol. 13, no. 8, p.186, 2021.

R.Prasad, V.Rohokale, R.Prasad, and V.Rohokale, "Artificial intelligence and machine learning in cyber security," Cyber security: the lifeline of information and communication technology, pp.231-247, 2020.

T.C.Truong, I.Zelinka, J.Plucar,M.Čandík, and V.Šulc,"Artificial intelligence and cybersecurity: Past, presence, and future," In Artificial intelligence and evolutionary computations in engineering systems, pp. 351-363, Springer Singapore, 2020.

I.H.Sarker, Y.B.Abushark, F.Alsolami, and A.I. Khan,"Intrudtree: a machine learning based cyber security intrusion detection model," Symmetry, vol. 12, no. 5, p.754, 2020.

Diro, and N.Chilamkurti, "Distributed attack detection scheme using deep learning

approach for Internet of Things," Future Generation Computer Systems, vol. 82, pp.761-768, 2018.

Z.Zhang, H.Ning, F.Shi, F.Farha, Y.Xu, F.Zhang, and K.K.R. Choo,"Artificial intelligence in cyber security:

research advances, challenges, and opportunities," Artificial Intelligence Review, pp.1-25, 2022.

I.F.Kilincer, F.Ertam, and A.Sengur, "Machine learning methods for cyber security intrusion detection:

Datasets and comparative study," Computer Networks, vol. 188, p.107840, 2021.

Ramkumar, M. S., Emayavaramban, G., Amudha, A., Nagaveni, P., Divyapriya, S., & SivaramKrishnan, M. (2021, October). A Hybrid AI Based and IoT Model Generation of Nonconventional Resource of Energy. In 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC) (pp. 1-6). IEEE.

Network Intrusion Detection. Available online: https://www.kaggle.com/ (accessed on 12 March 2020).

H. Alqahtani, I.H. Sarker, A. Kalim, S.M. Minhaz Hossain, S. Ikhlaq, and S. Hossain, 2020. "Cyber intrusion detection using machine learning classification techniques", In Computing Science, Communication and Security:

First International Conference, COMS2 2020, Gujarat, India, March 26–27, 2020, Revised Selected Papers 1 (pp. 121-131). Springer Singapore.

Kumar, A. Senthil, and EaswaranIyer. "An industrial iot in engineering and manufacturing industries— benefits and challenges." International journal of mechanical and production engineering research and dvelopment (IJMPERD) 9.2 (2019): 151-160