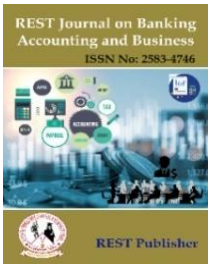
REST Journal on Banking, Accounting and Business

Vol: 4(4), December 2025

REST Publisher; ISSN: 2583 4746

Website: http://restpublisher.com/journals/jbab/

DOI: https://doi.org/10.46632/jbab/4/4/5



Enhancing Cybersecurity-Predicting and Detecting Cyber Hacking Breaches Using Machine Learning and Random Forest Classifier

S. Chandra Sekaran, K. Nandini, K.Sharmila, *A.Pooja

P.S.V College of Engineering and Technology, Krishnagiri, Tamil Nadu, India.

*Corresponding author Email: Poojabe22@gmail.com

Abstract. *This comprehensive literature review explores the transformative potential of artificial intelligence in strengthening cybersecurity defenses against sophisticated digital threats that circumvent traditional security measures. Grounded in cybersecurity-deep learning theory and AI pattern recognition principles, this study systematically analyzes how AI-driven technologies enhance threat detection, predictive vulnerability assessment, and insider threat analysis through behavioral insights.The findings emphasize that effective cybersecurity advancement requires a balanced approach integrating AI's analytical capabilities with human expertise, supported by robust frameworks encompassing privacy safeguards, bias mitigation strategies, and explainable AI practices.*

1. INTRODUCTION

Traditional cybersecurity frameworks, which primarily rely on rule-based systems and signature detection methods, have proven insufficient in addressing the dynamic and unpredictable nature of modern threats that are evolving with significant speed and complexity. This critical gap in defensive capabilities has positioned artificial intelligence as a transformative force in cybersecurity. It leverages its ability for real-time data analysis, advanced pattern recognition, and predictive modeling to overcome the limitations of conventional approaches. Unlike static security solutions that depend on predefined parameters, AI-powered systems—particularly those employing machine learning can process vast amounts of data to identify subtle anomalies and emerging threats by learning from historical patterns and behavioral indicators, rather than solely relying on known threat signatures. These machine learning capabilities enable the detection of minute variations in network traffic, user activities, and file attributes that may indicate potential security breaches, and provide a more adaptive and predictive security mechanism essential for protecting increasingly complex digital ecosystems against the sophisticated tactics employed by contemporary cyber adversaries.

2. LITERATURE REVIEW

1. A Review On The Evaluation Of Feature Selection Using Machine Learning For Cyber- Attack Detection In Smart Grid.

This comprehensive analysis explores the multifaceted security challenges facing smart grid infrastructure by examining system-level vulnerabilities across all grid components and evaluating the performance of various cyber-attack detection methods, including rule-based systems, signature-based identification, anomaly detection techniques, and machine learning algorithms offering varying levels of threat recognition capabilities.

The research further investigates emerging cybersecurity solutions such as artificial intelligence-based approaches and blockchain technology, which promise enhanced security through adaptive learning and decentralized verification mechanisms, while acknowledging the persistent challenges and evolving threat landscape that characterize smart grid security.

Through a systematic analysis of both current vulnerabilities and future security strategies, this work provides policymakers

and industry stakeholders with the essential insights needed to develop robust, multi-layered security frameworks that can effectively protect critical energy infrastructure against increasingly sophisticated cyber threats, ensuring the continued reliability and security of modern power distribution systems that underpin economic activity and public welfare.

2.Cyber Attack Prediction: From Traditional Machine Learning To Generative Artificial Intelligence.

This comprehensive research explores how the strategic application of several advanced techniques, including artificial intelligence, machine learning for pattern recognition, deep learning for sophisticated threat analysis, natural language processing for processing textual threat intelligence, explainable AI for transparent decision-making processes, and generative AI for simulating attack scenarios and developing defensive strategies, operates.

This study makes a significant contribution by providing a comparative evaluation of machine learning and deep learning methods to assess their accuracy and applicability across various security challenges, its investigation into explainable AI approaches that enhance transparency in anomaly detection systems, and its exploration of cutting-edge generative AI and natural language processing techniques capable of generating threat intelligence and conducting sophisticated attack simulations.

By examining both the theoretical frameworks and real-world applications of generative AI in commercial cybersecurity products, this research provides critical insights for developing effective, reliable, and explainable AI-driven security solutions that can dynamically adapt to evolving threats. In an era where cyber resilience, operational continuity, and information security have become fundamental, this enhances the industry's ability to mitigate cyber risks and strengthen the overall security posture of digital infrastructures against increasingly complex adversarial activities.

3. METHODOLOGY

A. Existing System: Recognizing the multifaceted nature of these threat factors, understanding their operational characteristics, identifying their common indicators and attack methods, and comprehending the specific vulnerabilities they exploit are fundamental to developing effective cybersecurity frameworks that can anticipate, detect, and neutralize threats before they cause significant damage to an organization's assets, critical data, and essential infrastructure systems.

1.Malware Attacks: These threats typically infiltrate systems through multiple attack vectors including infected email attachments that exploit user trust, compromised websites that deliver drive-by downloads, malicious software distributed through unofficial channels, exploited vulnerabilities in outdated applications, and social engineering tactics that deceive users into executing harmful code.

2.Phishing Attacks: These attacks exploit human psychology rather than technical vulnerabilities, using trust, authority, fear, and urgency to bypass security controls, making user awareness and verification protocols essential defensive measures. Along with technical solutions such as email filtering, multi-factor authentication, and anti-phishing technologies, they detect and block fraudulent communications before they reach potential victims, preventing financial losses or data breaches.

B.Proposed System: Advancing AI-driven cybersecurity requires extensive research in establishing an optimal balance between automated systems and human oversight, as current findings demonstrate AI's powerful capabilities in real-time data processing and predictive threat detection, while also revealing limitations in adapting to complex operational environments that necessitate contextual judgment and human expertise.To ensure technology augments rather than displaces professional skills, future research should prioritize developing hybrid models that integrate AI automation with human decision-making, thereby addressing critical concerns about skill erosion among cybersecurity practitioners who may become overly reliant on automated systems.Additionally, substantial inquiry into ethical implications and privacy considerations is essential, particularly regarding AI's pervasive surveillance capabilities, which risk inadvertently infringing upon individual privacy rights, necessitating transparent deployment frameworks and accountability mechanisms, including auditing protocols for evaluating "black-box" AI models whose opaque decision-making processes complicate attribution of responsibility.This research also highlights significant disparities in AI resource access, particularly affecting small and medium-sized businesses constrained by budgetary limitations that hinder the adoption of advanced security solutions.Future studies should explore collaborative approaches such as shared AI platforms, cybersecurity consortia, and multi-stakeholder partnerships that implement resource-pooled security

strategies to mitigate the growing security gap between well-funded corporations and resource-constrained organizations.

4. DATASET

In cybersecurity contexts, datasets may include network traffic logs capturing packet information, user behavior logs documenting access patterns and authentication attempts, malware samples with associated attributes and signatures, threat intelligence feeds containing indicators of compromise, or system performance metrics reflecting normal and anomalous operational states. The scope of a dataset can range from a single database table containing thousands or millions of records for training supervised learning algorithms, to more complex collections of multiple interconnected tables related to extensive test observations or real-world events, providing the rich, multidimensional information necessary to build robust machine learning models capable of detecting sophisticated cyber threats, predicting vulnerabilities, classifying malicious activities, and adapting to evolving attack patterns through continuous learning from historical and contemporary security data.

IMPLEMENTATION:

A. File Design: The relational database model offers significant advantages by organizing information in multiple interconnected tables that maintain logical relationships through primary and foreign key constraints, enabling sophisticated data operations such as joins, unions, and built-in queries that can combine, aggregate, and present information across related entities, while simultaneously protecting data integrity and reducing redundancy through normalization principles.

- Across networks
- Via the internet
- Through laptops and other electronic devices
- With other software systems

Implementing the relevant database structure in the "Calorie Prediction using Deep Learning" system represents a fundamental design decision that directly determines the system's performance, efficiency, and maintainability through careful optimization of file structures and strategic elimination of data redundancy, while ensuring comprehensive information capture for each entity within the application domain.

B. Input design: These safeguards contribute to several strategic objectives, including reducing data entry errors through clear interface design and immediate feedback, preventing erroneous data from corrupting the database through comprehensive validation rules, minimizing downstream processing errors and associated correction costs, enhancing the user experience by providing intuitive input forms with helpful instructions and error messages, maintaining the data quality standards required for machine learning model accuracy, and ultimately ensuring system reliability by establishing data integrity at the foundation where data enters the information processing environment.

- Creating cost-effective input
- High level of accuracy
- Unambiguous
- Data entry
- Data validation
- Sending data to the system
- Data correction

The primary objective of input design is to make data entry simple, logical, and error-free. This helps in reducing processing delays and improving the accuracy of computer outputs

C.External: In systems such as Online Ordering Systems or object-oriented applications, external inputs manifest through two primary categories: user-generated inputs where customers interact with the system by entering authentication credentials, searching for products, selecting items, providing delivery addresses, submitting payment information, and issuing commands that define their specific needs and expectations; and administrator-generated inputs where system managers perform control functions including updating product catalogs, managing user accounts, verifying transactions, configuring system parameters, monitoring operational activities, and maintaining database integrity to ensure accurate and secure system operation.

D.Internal: Input formats and structures define the structured specifications and conventions that interpret external data, requiring information to be presented in precise arrangements such as text strings, numerical values, image files, or command sequences. These enable accurate parsing and processing by the application's internal components.

When users interact with a computer through keyboards, mouse clicks, touchscreens, or file uploads, their actions generate raw input data that passes through a window or graphical user interface system. This system translates these physical interactions into standardized messages or data structures that the underlying software can recognize and interpret according to predefined protocols.

These structured messages, based on established rules, route them to appropriate processing modules, triggering specific functions such as form submissions, database queries, authentication procedures, or computational tasks corresponding to the user's intended action.

The systematic approach to building such systems is governed by a development methodology – a comprehensive framework, technique, or algorithmic strategy that guides the entire system design, implementation, testing, and maintenance lifecycle – with various methodologies including the Waterfall's sequential phase progression suitable for projects with stable requirements, Agile's iterative development accommodating frequent changes and continuous user feedback, the Spiral's risk-driven approach for complex systems, and the evolutionary refinement of prototypes through user interaction.

5. TESING

The testing process evaluates several key attributes, including functional correctness to ensure the system performs the required functions accurately, performance efficiency to verify acceptable response times and resource utilization, security robustness to identify vulnerabilities that could be exploited by malicious actors, usability to ensure intuitive user interactions, compatibility across different platforms and environments, reliability to assess consistent operation under various conditions, and maintainability to evaluate the ease of future modifications and updates.

- It meets the requirements that guide its design and development.
- It responds correctly to all types of inputs.
- It performs its functions within an acceptable timeframe.
- It is sufficiently usable.
- It can be installed and operated in the intended environments.
- It achieves the desired outcome for its stakeholders.

A. Unit Testing: This white-box testing approach is performed by software developers during the construction phase of the development life cycle, where they verify that each building block functions according to its specifications before integrating it with other components. Often, a single function requires multiple test cases to thoroughly explore different execution paths, boundary conditions, edge cases, error handling procedures, and corner scenarios that can reveal flaws in the logic or implementation. While unit testing alone cannot guarantee the functionality of the overall system or verify the complex interactions between components, it serves as an essential defect prevention and early detection strategy, identifying construction errors, logical flaws, and coding mistakes at their source, before defects become exponentially costlier and time-consuming to diagnose and fix in code integration testing, system testing, or production environments.

B. Integration Testing: The testing process progressively expands the scope of integration, starting with closely related component pairs or small clusters, then progressing to subsystem-level integration combining multiple modules, and finally culminating in a fully functional system-level test where all structural components are integrated, ensuring that data flows correctly through the data processing pipelines, control logic components are

properly integrated, and the system delivers the intended end-to-end functionality. In security-critical applications such as cybersecurity systems, integration testing is crucial to ensure that threat detection modules interface correctly with logging systems, alerting mechanisms trigger response workflows correctly, authentication components securely interact with access control systems, and data sharing between analytical tools maintains confidentiality and integrity across distributed processing frameworks.

C.Functional Testing: This testing method focuses on answering fundamental questions such as, "Can users successfully complete this task?" and "Does this specific feature function as designed?" by systematically examining the system from the end-user's perspective, without concern for internal code structure or implementation details. It is a black-box testing technique that evaluates inputs, outputs, and system behavior against predefined acceptance criteria.
Correct Input: All identified types of valid input data should be accepted by the system and processed correctly, producing the expected results without errors.

Invalid Input: All identified types of invalid input data should be rejected by the system, and appropriate error messages or validations should be triggered to prevent incorrect processing.

Functionality: All identified functionalities or features of the application should be executed during testing to verify that they perform as intended according to the requirements.

Output: All identified types of output from the application should be generated and verified to ensure their correctness, completeness, and consistency with the expected results.

System Testing: System testing involves testing a fully integrated system to verify that it meets specified requirements.

Performance Testing: erformance testing evaluates the speed, responsiveness, and stability of the system under expected workloads. It ensures that the system produces result within specified time limits and meets performance requirements. Performance testing is categorized under black-box testing.

D.White-Box Testing: This methodology employs various formal techniques, including statement coverage, which ensures that every executable line is tested; branch coverage, which ensures that all conditional paths are executed; path testing, which explores different path combinations through the code; control flow testing, which verifies the correct program execution sequences; data flow testing, which monitors variable states and changes throughout the execution; and decision coverage, which ensures that all Boolean expressions are evaluated under both true and false conditions.

White-Box Testing Techniques
1) Tata Blow Testing
2) Kondol Impact Test
3) French Kaviraj Test
4) Report Kaviraj Test
5) Research Report Kovaraj Test

E. Branch coverage testing: This approach focuses on comprehensive testing of decision structures, including if-else statements, switch-case constructs, loop conditions, and ternary operators, creating test scenarios that force execution through every alternative branch, effectively exposing improperly implemented conditional logic that might otherwise go undetected with logical errors, unreachable code sections, or less thorough testing approaches. While branch coverage and decision coverage techniques share conceptual similarities in their focus on validating conditional logic, there is a subtle but crucial difference: decision coverage ensures that each decision point evaluates to both true and false at least once, whereas branch coverage more rigorously demands that every individual branch emanating from those decision points—including compound conditions with multiple logical operators—must be traversed independently during the testing process.

Statement Coverage Testing: Statement coverage is a very basic and widely accepted white-box testing technique. It measures the completeness of testing by ensuring that every executable statement in the source code is executed at least once during the testing process. It calculates coverage by dividing the number of executed statements by the total number of statements and provides metrics for developers to assess the completeness of their test suites.

H. Greyson Testing: This methodology focuses on testing across all architectural layers, including presentation interfaces, application logic layers, data access layers, and backend systems, thereby increasing overall test coverage and effectiveness by simultaneously verifying both user-facing functionality and internal processing accuracy. Gray box testing views the primary application in integration testing scenarios, where understanding component interfaces enhances test design, and also in penetration testing exercises where security professionals utilize limited system knowledge to simulate informed attackers, enabling more realistic assessments of vulnerabilities that could be exploited by adversaries possessing internal knowledge, social engineering insights, or information gathered through initial attacks, making it an essential technique for comprehensive security verification in modern cybersecurity strategies.

6. SYSTEM ARCHITECTURE

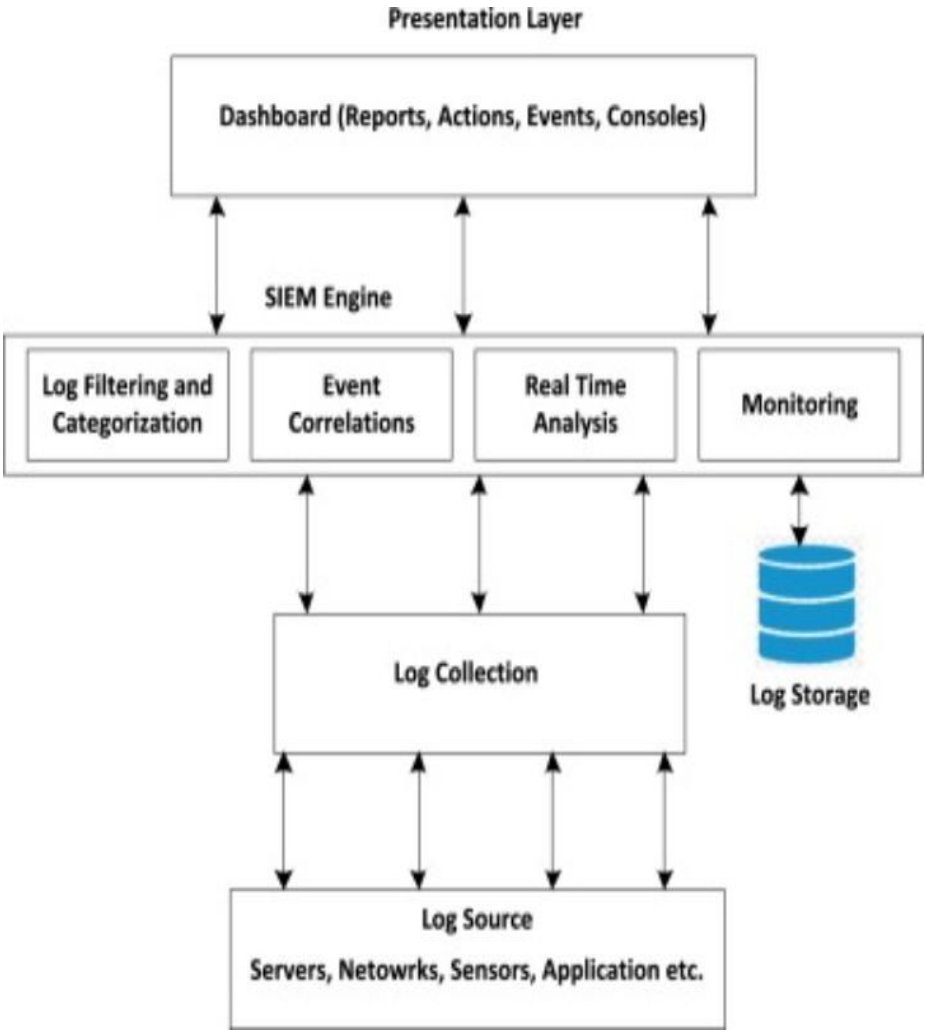


FIGURE 1. Presentation Layer

7. CONCLUSION

This comprehensive study explored the transformative potential of artificial intelligence in enhancing cybersecurity capabilities through improved threat detection accuracy, predictive vulnerability assessment, and accelerated incident response mechanisms, addressing the critical imperative for adaptive security frameworks capable of confronting sophisticated and rapidly evolving digital threats that challenge conventional security strategies. Utilizing an integrated literature review methodology to synthesize insights from contemporary research on AI applications in cybersecurity, this investigation generated novel knowledge that combines theoretical understanding with practical implementation guidance for policymakers and security practitioners navigating the complex integration of intelligent technologies into organizational security frameworks.

REFERENCE

[1]. M. V. Manoj Kumar ,K. Ashoka, M. S. Abdul Razak ,Andr.Naseer. A Comprehensive Survey On *Cognitive Cyber Security Analysis Using Machine Learning Approaches . 21 August 2025.*

[2]. *Mohammed Ashfaaq Deepthi, N. Ratnayake. Ai-Powered System For An Efficient And Effective Cyber Incidents Detection And Response In Cloud Environments. 14 April 2024.*

[3]. *Dhiya Al-Jumeily. A Review On The Evaluation Of Feature Selection Using Machine Learning For Cyber-Attack Detection In Smart Grid. 10 February 2024.*

[4]. *ShilpaAnkalakiGeetabaiSHukkeri,AparnarajeshAtmakuri,M.Pallavi*

[5]. *,Tonyjan ,And Ganeshr Naik. Cyber Attack Prediction: From Traditional Machine Learning To Generative Artificial Intelligence. 7 February 2025.*

[6]. *Dr Nandini N, Pooja M S. Cyber Hacking Breaches Prediction And Detection Using Machine Learning. 8 August 2025.*

[7]. *Ganesh S. Nayak, Balachandra Muniyal. Enhancing Phishing Detection: A Machine Learning Approach With Feature Selection And Deep Learning Models. 14 January 2025.*

[8]. *Jia Yu ,Alexey V. Shvetsov, And Saeed Hamood. Leveraging Machine LearningForCyberSecurityResilience InIndustry4.0: ChallengesAndFuture Directions.27 September 2024.*

[9]. *Naveen Pakalapati ,Muthukrishnan Muthusubramanian.Machine Learning For Cyber Security Threat Detection And Prevention. 2, February2024.*

[10]. *Yousik Lee Donghoonlee,Samuelwoo,Yunkeuns. Practical Vulnerability- Information-Sharing Architecture For Automotive Security-Risk Analysis. 27 May 2020.*