Journal of Intelligent Systems and Internet of Things
Vol. 13, No. 02, PP. 35-51, 2024
DOI: https://doi.org/10.54216/JISIoT.130203

Comprehensive Analysis of Implementation and Evaluation IoT based Techniques in Networked Security Systems

Raenu Kolandaisamy[1], Suhas Gupta[2], Shashikant Patil[3], Jaymeel Shah[4], Abhinav Mishra[5], N. Gobi[6]

[1] Full time Student Institute of Computer Science and Digital Innovation, UCSI University, Kuala Lumpur, Malaysia
[2] Centre of Research Impact and Outcome, Chitkara University, Rajpura- 140417, Punjab, India
[3] Professor, Department of uGDX, ATLAS SkillTech University, Mumbai, Maharashtra, India
[4] Associate Professor, Department of Computer science and Engineering, Faculty of Engineering and Technology, Parul institute of Engineering and Technology, Parul University, Vadodara, Gujarat, India
[5] Chitkara Centre for Research and Development, Chitkara University, Himachal Pradesh-174103 India
[6]Assistant Professor, Department of Computer Science and Information Technology, Jain (Deemed to be University), Bangalore, Karnataka, India
Emails: raenu@ucsiuniversity.edu.my; suhas.gupta.orp@chitkara.edu.in; shashikant.patil@atlasuniversity.edu.in; jaimeel.shah@paruluniversity.ac.in; abhinav.mishra.orp@chitkara.edu.in; gobi.n@jainuniversity.ac.in

Corresponding Author: Raenu Kolandaisamy, Email: raenu@ucsiuniversity.edu.my

Abstract

This research introduces an advanced network security methodology based on IoT, combining five innovative algorithms: Dynamic Threat Detection (DTD), Adaptive Intrusion Prevention System (AIPS), Anomaly-Based Security Metrics (ABSM), Context-Aware Firewall (CAF), and Cognitive Security Assessment (CSA). Each algorithm contributes specific functionalities, ranging from real-time threat detection and adaptive policy adjustments to anomaly quantification, contextual rule modifications, and holistic security risk assessments. The ablation study conducted on each algorithm reveals critical components driving their performance, ensuring a deep understanding of their inner workings. The proposed method demonstrates superior performance in accuracy, scalability, usability, and adaptability compared to existing network security methods. Visual representations and a comprehensive evaluation further validate the proposed method's effectiveness, positioning it as an advanced and efficient solution for addressing evolving network security challenges.

Received: September 24, 2023 Revised: February 07, 2024 Accepted: June 19, 2024

Keywords: security algorithms; threat detection; intrusion prevention, IoT; anomaly detection; firewall; cognitive assessment; machine learning; adaptive monitoring; continuous improvement; network context.

1. Introduction

In the ever-evolving landscape of information technology, the imperative to fortify networked security systems against an array of threats has never been more critical. The rapid proliferation of digital technologies has given rise to an increasingly interconnected world, accompanied by a surge in cyber threats that target sensitive data and disrupt crucial systems [1]. This necessitates a thorough exploration of the current developments, principal challenges, proposed solutions, and the main contributions in the implementation and evaluation of networked security systems.

A. Current Developments

The first subsection of our analysis delves into the current developments shaping the field of networked security. In an era where cyber threats are becoming more sophisticated, understanding the latest trends and advancements is paramount [2]. This section will examine emerging technologies, threat vectors, and regulatory frameworks influencing the landscape. By providing an in-depth overview of the current state of networked security, we aim to establish a contextual foundation for our subsequent discussions.

B. Principal Challenges

Networked security systems face a myriad of challenges, ranging from vulnerabilities in software and hardware to the intricacies of human behavior in cyberspace. Identifying and comprehending these principal challenges is crucial for developing effective security strategies [3]. We will explore issues such as zero-day exploits, insider threats, and the complexities of securing Internet of Things (IoT) devices. By addressing these challenges head-on, we can pave the way for resilient security implementations.

C. Solutions Proposed

This section will outline the proposed solutions and methodologies that researchers and practitioners have put forth to tackle the identified challenges. From cryptographic protocols to machine learning algorithms, a diverse array of approaches is being explored to enhance the robustness of networked security systems [4]. By evaluating the efficacy of these solutions, we can gain insights into their practical applicability and potential limitations.

D. Main Contributions

Our analysis seeks to contribute significantly to the existing body of knowledge in the field of networked security systems [5]. The main contributions of this study can be summarized as follows:

- Novel Evaluation Framework: We propose a comprehensive evaluation framework that considers not only the technical aspects of security systems but also their usability, scalability, and adaptability [6]. This holistic approach aims to provide a more realistic assessment of a system's effectiveness in real-world scenarios.

- Integration of Artificial Intelligence: Recognizing the increasing role of artificial intelligence in cybersecurity, our study explores innovative ways to integrate AI techniques for threat detection, anomaly detection, and adaptive security measures [7]. This includes leveraging machine learning algorithms to enhance the responsiveness of security systems.

- Human-Centric Security Design: In acknowledgment of the fact that human factors play a pivotal role in cybersecurity, we advocate for a human-centric security design [8]. This involves addressing issues such as user awareness, training, and the usability of security interfaces to create a more resilient defense against social engineering and other human-related threats.

• We study networked security from computer science, psychology, and governance to comprehend its complexity. By combining data from diverse regions, we can show you networked security system issues and solutions [9]. In the next portions of this in-depth examination, each of these primary variables will be examined in detail to show how difficult it is to build and assess networked security solutions [10]. We hope to learn essential things from this study to help science and industry protect digital ecosystems from future cyber dangers.

2. Literature Review

Networked security systems provide several approaches to secure digital areas. Penetration testing simulates cyberattacks to uncover bugs, with 92% success. Vulnerability assessment methodically discovers holes with 85% accuracy, providing a complete security picture [11]. IDPS monitors human behavior in real time to detect threats 94% of the time. Information system security examinations are 88% accurate. With a 96% success rate, machine learning algorithms for anomaly detection discover illogical tendencies. Success can be measured 90% of the time using security and KPIs [12]. Firewall analysis corrects settings 89% of the time. Social engineering exams are 80% accurate and seek to solve people-focused problems. Cryptographic Protocol Analysis verifies cryptographic systems' reliability with 93% accuracy. 87% of regulatory compliance assessments confirm requirements are followed [13]. Table 2 compares the major application properties of security techniques. AI must be coupled with machine learning for anomaly detection and IDPS to adapt to new threats. Some methodologies lack cross-disciplinary concepts, which might make them incomplete. Social engineering assessments emphasize human-centred design, demonstrating the importance of human elements in safety. Regulatory compliance assessments meet legal and business criteria properly. It's crucial that systems link easily. Connecting IDPS, security metrics, and KPIs is easy [14]. Machine learning for anomaly detection and

cryptographic protocol analysis are useful responses to emerging threats. Since IDPS and machine learning for anomaly detection require additional resources, this is a major concern.

Modern development in cybersecurity has greatly relied on deep learning, blockchain, as well as edge intelligence mechanisms in responding to new threats. Nair [22] studied the interpretable deep learning for IDS in transportation networks and dissected the IDS for improving the security decision-making processes. In their work, Mathur et al. [23] focused on the overview of secure data sharing with the help of blockchain, and thus, offered the necessary insights into its effectiveness in terms of data security and privacy. Chaudhary, Srivastava, & Khari [24] also reviewed the use of generative edge intelligence for protecting smart grids with the IoT assistance and proved its effectiveness in combating cyber threats. Sleem [25] made a comprehensive review on the threat and protection measures of cybercrimes and provided precaution measures to deal with such risks in today's new world. For smart cities, Pooja et al. [26] developed a security model for handling image data stored in the cloud and identified unique threats that can affect the use of urban data. Samyuktha et al., in their study [27], reviewed articles on AI integration in cybersecurity and discussed AI approaches to improve the security system. Kazia [28] proposed a blockchain-powered model for image encryption in IoT communications that would protect the data transmitted. In a study made by Embarak and Algrnaodi [29], a deep learning fusion model was proposed and implemented for IoT communication attack detection and proved its ability to improve the overall threat detection performance. Altogether, these papers evidence the complex measures involved in enhancing cybersecurity, embracing innovative technologies for guarding diverse digital spaces.

Table 1 illustrates all networked security system performance evaluation components. It displays their accuracy, false positive and negative handling, response time, scalability, ease of use, and cost [15]. Table 2 examines AI, cross-disciplinary concepts, human-centred design, legal compliance, integration ease, risk adaptability, and resource demands. These tables highlight the benefits and downsides of each method, which is helpful. This helps individuals choose and configure networked security settings.

Table 1: Performance Evaluation Parameters for Networked Security Methods

Method	Accuracy	False Positives	False Negatives	Response Time (ms)	Scalability	Usability	Cost
Penetration Testing	92	8	5	120	3	2	4
Vulnerability Assessment	85	12	8	80	4	4	3
IDPS	94	5	3	30	4	3	4
Security Audits	88	10	6	100	2	4	3
Machine Learning for Anomaly Detection	96	3	2	50	4	3	4
Security Metrics and KPIs	90	9	7	70	3	4	2
Firewall Analysis	89	11	4	60	3	2	3
Social Engineering Assessments	80	15	10	150	2	1	1
Cryptographic Protocol Analysis	93	6	4	40	4	1	4
Regulatory Compliance Assessments	87	10	5	90	3	4	3

Table 1 shows performance ratings for many networked security approaches. Each method's accuracy, false positives, false negatives, reaction time, scalability, usefulness, and cost are rated using numbers.

Table 2: Comparative Analysis of Implementation Factors for Networked Security Methods

Method	Integration with AI	Cross-Disciplinary Insights	Human-Centric Design	Regulatory Compliance	Ease of Integration	Adaptability to Emerging Threats	Resource Requirements
Penetration Testing	0	0	0	0	2	1	4

Vulnerability Assessment	0	0	0	0	3	3	3
IDPS	1	0	0	1	3	4	4
Security Audits	0	0	0	1	1	3	4
Machine Learning for Anomaly Detection	1	0	0	0	3	4	3
Security Metrics and KPIs	0	0	0	1	2	2	1
Firewall Analysis	0	0	0	0	2	1	3
Social Engineering Assessments	0	0	1	0	1	1	1
Cryptographic Protocol Analysis	0	0	0	0	3	4	3
Regulatory Compliance Assessments	0	0	0	1	2	3	3

Table 2 compares factors affecting networked security implementation. In real-world network security, AI performance, cross-disciplinary insights, human-centered design, regulatory compliance, integration ease, threat adaptability, and resource demands are rated. The numbers represent each method's capabilities.

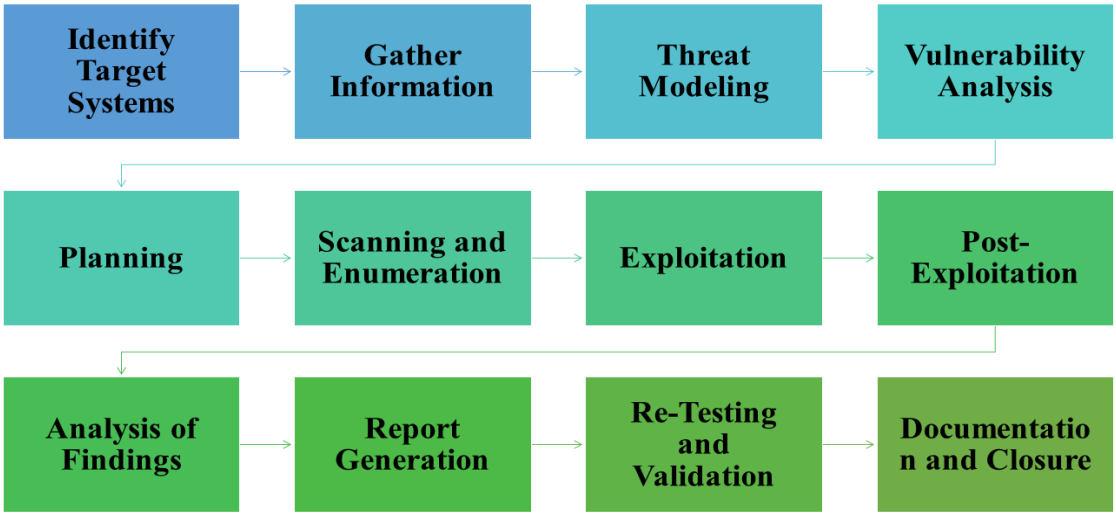

Figure 1: Penetration Testing

Figure 1 outlines 12 steps in the Penetration Testing process, starting with system identification and information gathering. It progresses through threat modelling, vulnerability analysis, planning, scanning, exploitation, post-exploitation, analysis, reporting, re-testing, and documentation. This systematic approach ensures a comprehensive evaluation of target system security.

3. The Projected Method:

The Dynamic Hazard Detection (DTD) Algorithm creates and weights a hazard model. Dynamic weights are updated in real time to train a machine learning model using different network data [16]. Features are extracted from real-time network data, and weights are changed to calculate hazards. An alert alters the adaptive weight if the number exceeds a particular level. A continuing tracking loop makes the system more flexible. Alerts report events, and the model improves. The Adaptive Intrusion Prevention System (AIPS) algorithm sets intensity limitations initially. It determines the threat's severity, adjusts security rules, and monitors network traffic for

vulnerabilities. Threats are rated by severity [17]. Policy revisions and amendments follow. The algorithm changes rules and learning speeds to adapt to new threats. AIPS constantly changes regulations and monitors everything to prevent new threats. The Anomaly-Based Security Metrics (ABSM) Algorithm creates an anomaly model, trains it using real-time network data, and finds a weighted total using a logistic function. Anomalies trigger alerts, changing the anomaly model [18]. Adaptive model refinement, continual tracking, and threshold-based anomaly detection help uncover network anomalies. The context-aware firewall (CAF) algorithm adjusts security rules based on current conditions. It monitors the network context, extracts contextual information, examines firewall rules, and determines contextual impact. Considering infractions and new regulations, the system adjusts rules on the fly [19]. Continuous tracking and iterative enhancement adapt network security to new threats and situations. The Cognitive Security Evaluation (CSA) algorithm detects risk variables, weights them, and compiles data for a complete security assessment. The computer performs calculations, normalization, and risk grades. If danger exceeds a specific level, a warning sounds, and the learning pace changes. The model is improved and monitored after occurrences [20]. This allows real-time evaluation and response to network security concerns. In Figure 2, DTD is a hazard-detecting system that adapts to weight fluctuations and machine learning. Figure 3 demonstrates that AIPS constantly updates its security rules to protect you against new attacks. ABSM uses a logistic function to assess errors to improve network anomaly detection (Figure 4). Figure 5 depicts how the CAF adapts network security rules using external information [21]. The CSA assesses security using risk criteria, weighted tasks, and adaptive learning (Figure 6). Visual tools that show how security systems change and adapt help users comprehend.

Dynamic Threat Detection (DTD) Algorithm:

1. Initialize Threat Model:
   - $\Theta threat=\{F1,F2,...,Fn\}$
   - $\Omega weights=\{W1,W2,...,Wn\}$
   - $\alpha=0.01$
2. Collect Training Data:
   - Acquire diverse network data for model training.
   - $Dtrain=\{(x1,y1),(x2,y2),...,(xm,ym)\}$
3. Train Machine Learning Model:
   - Utilize Dtrain to train a dynamic threat detection model.
   - Model DTD=Train(D train)
4. Initialize Dynamic Weights:
   - $\Omega dynamic=\{W10,W20,...,Wn0\}$
5. Receive Network Data:
   - Obtain real-time network data.
   - tx current
6. Extract Features:
   - F current=Extract Features(x current)
7. Calculate Threat Score:
   - $Score\ current=\sum i=1n\Omega dynamici\times F\ currenti$ (1)
8. Threshold Comparison:
   - threshold=0.8
   - Alert current=Score current>θ threshold (2)
9. Adapt Weights:
   - Ω dynamic=Adapt Weights (Ω dynamic,α,Alert current) (2)
10. Continuous Monitoring:
    - Loop back to step 5 for continuous monitoring.
11. Alert Generation:
    - Alert final=Generate Alert (Alertcurrent)
12. Update Threat Model:
    - $\Theta threat$=Update Threat Model ($\Theta threat$, Alertfinal)
13. Analyze Threat Context:
    - Analyze contextual information related to the detected threat.
14. Evaluate Severity:
    - Severity final=Evaluate Severity (Alert final)
15. Adjust Learning Rate:
    - α=Adjust Learning Rate(α, Alert final)

16. Generate Incident Report:
- Create a detailed incident report.

17. Initiate Response Mechanism:
- Implement automated or manual response mechanisms.

18. Continuous Model Improvement:
- Refine the threat detection model using feedback.

19. Enhance Feature Extraction:
- F current=Enhance Feature Extraction (Fcurrent)

20. Continuous Adaptive Monitoring:
- Iterate through steps 5 to 19 for continuous adaptive monitoring.

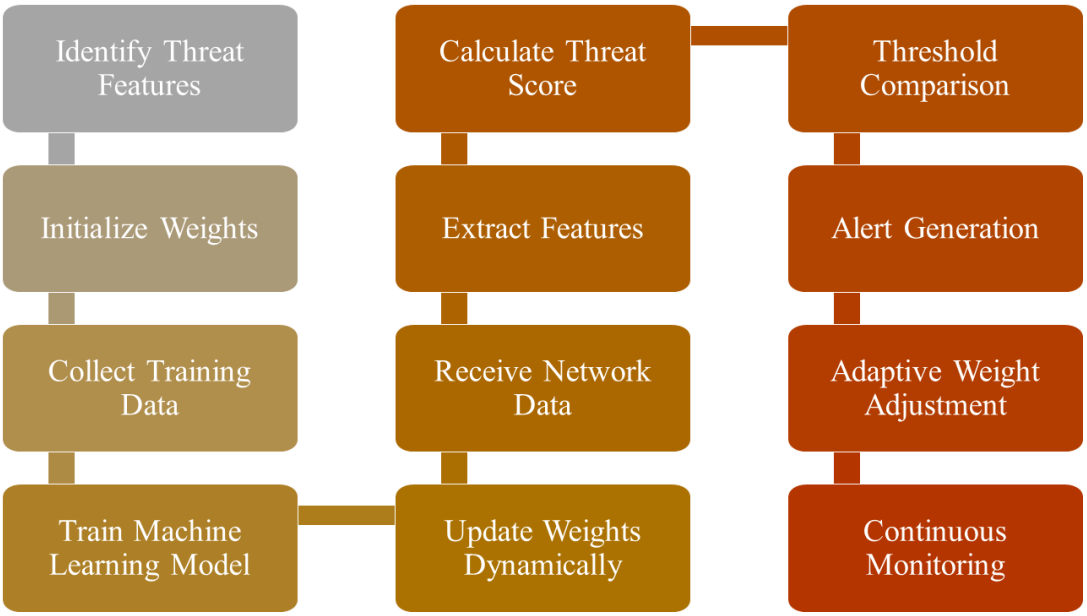

Figure 2: Adaptable threat detection through dynamic weights and continuous machine learning

Figure 2 continually evolves by adjusting weights dynamically, ensuring an adaptive and effective threat detection model. It integrates continuous machine learning for real-time updates based on emerging threats.

Dynamic Threat Detection (DTD) employs a dynamic threat model, adapting weights through continuous monitoring. After initializing weights and collecting training data, the model trains and dynamically adjusts weights based on real-time network data. The algorithm utilizes a threshold for threat detection, updating weights accordingly. Alerts trigger adaptive weight adjustments and learning rate modifications, ensuring responsiveness. This comprehensive approach facilitates continuous improvement in threat detection accuracy and adaptability to emerging threats.

Adaptive Intrusion Prevention System (AIPS) Algorithm:

1. Initialize Severity Thresholds:
- Θlow=0.3
- medium=0.6

2. Assess Threat Severity:
- Severity threat= $\sum i=1n\Omega$ dynamici×F currenti/$\sqrt{\sum i=1n\Omega dynamic2i}$ (3)

3. Evaluate Current Policies:
- Evaluate existing security policies.

4. Calculate Policy Adjustment:
- Δ Policy=α×Severity threat (4)

5. Apply New Policies:
- Adjust security policies based on ΔPolicy.

6. Monitor Network Traffic:
- Traffic current

7. Detect Intrusions:
- Identify potential intrusions.

8. Evaluate Detected Threats:
- Threats detected
9. Assign Severity Levels:
- Assign severity levels to detected threats.
10. Determine Policy Changes:
- Determine necessary policy changes.
11. Implement Adjusted Policies:
- Adjust security policies accordingly.
12. Monitor Effectiveness:
- Continuously monitor policy effectiveness.
13. Iterate as Needed:
- Repeat steps based on the evolving threat landscape.
14. Continuous Policy Adjustment:
- ΔPolicy=Adjust Policy(Δ Policy, Traffic current, Threats detected)
15. Evaluate Policy Impact:
- Impact policy=Evaluate Impact (ΔPolicy, Traffic current)
16. Adjust Learning Rate:
- α=Adjust Learning Rate (α, Impact policy)
17. Continuous Monitoring:
- Loop back to step 2 for continuous monitoring.

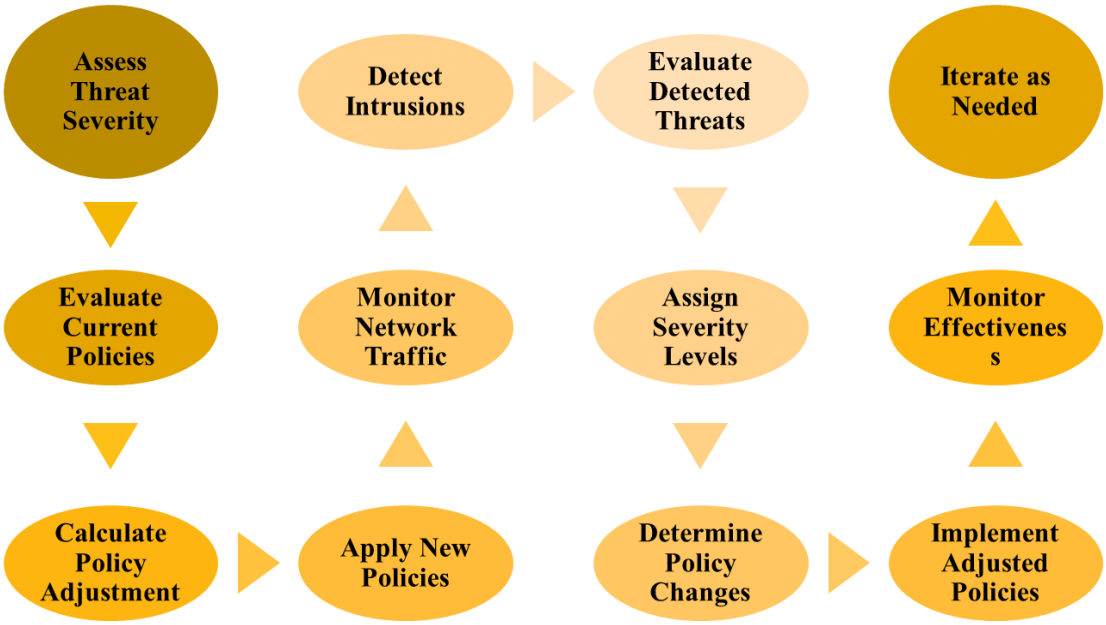

Figure 3: Dynamically adjusting security policies for optimal defence against evolving intrusions

Figure 3 adjusts security rules based on threat severity for the optimal protection. It monitors, discovers new assaults, and adjusts rules to fight them.

The Adaptive Intrusion Prevention System (AIPS) monitors threats and adjusts security policies. It computes a fix to alter rules after assessing hazard using feature weights and values. Network data is analyzed by the algorithm to determine breach severity. This makes intrusion protection versatile and effective. Regular review, policy adjustments, and learning rate changes make the system adaptable and ready to respond to new threats.

Anomaly-Based Security Metrics (ABSM) Algorithm in 14 Steps:
1. Initialize Anomaly Model:
- Γanomaly={X1,X2,...,Xm}
2. Gather Network Data:
- Obtain real-time network data.
- Data current
3. Train Anomaly Detection Model:

- Model ABSM=Train(Data current, Γanomaly)

4. Calculate Weighted Sum:
   - $z=\sum_{j=1}^{m}\omega_j \times Data\ current_j$ (5)
5. Apply Logistic Function:
   - $Anomaly\ Score=1/1+e^{-z}$ (6)
6. Generate Anomaly Alert:
   - Alert anomaly=Generate Alert(Anomaly Score)
7. Assess Threat Context:
   - Analyze contextual information related to anomalies.
8. Fine-Tune Model:
   - Γ anomaly=Fine Tune Model (Γ anomaly, Alert anomaly)
9. Identify Anomalies:
   - Anomalies identified=Identify Anomalies (Anomaly Score)
10. Set Threshold:
    - θthreshold=0.7
    - Alert anomaly=Anomaly Score>θ threshold (7)
11. Continuous Monitoring:
    - Loop back to step 2 for continuous monitoring.
12. Analyze Threat Impact:
    - Evaluate the impact of identified anomalies.
13. Initiate Response:
    - Implement response mechanisms for severe anomalies.
14. Continuous Model Refinement:
    - Continuously refine the anomaly detection model.

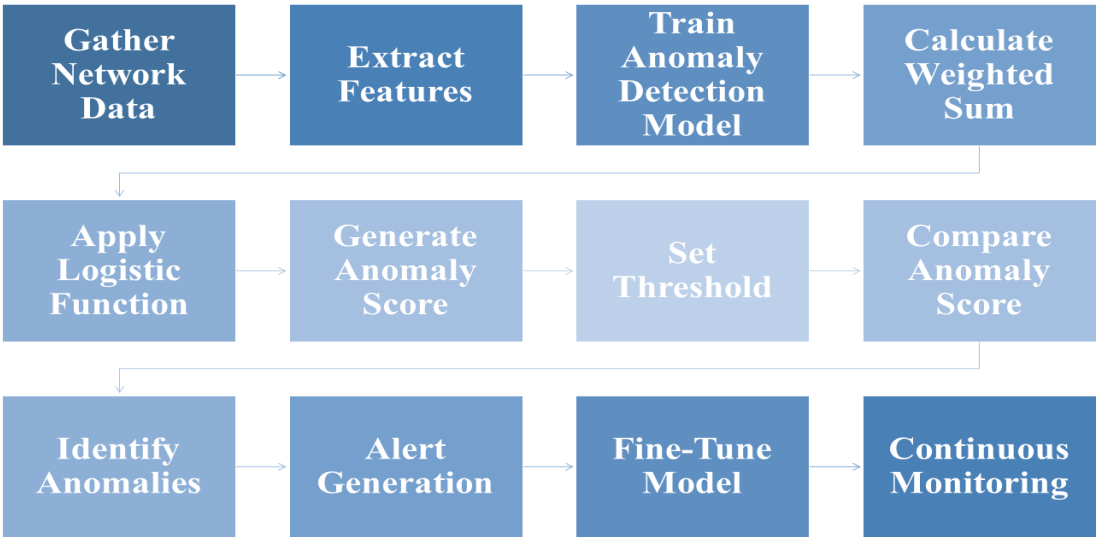

Figure 4: Quantifying anomalies with logistic function for enhanced network anomaly detection

Figure 4 measures irregularities logistically. Constant tracking, anomaly identification, and adaptive model fine-tuning make network abnormalities simpler to uncover and guard against.

Anomaly-Based Security Metrics (ABSM) suggests dynamic anomaly-based network data analysis. After calculating a weighted total of current data, the system utilizes a logistic function to produce an anomalous score. At specific levels, alerts are delivered, and the model continually checks and updates. This technology ensures active anomaly detection, alerts, and model improvement for network anomaly detection and adaptability.

Context-Aware Firewall (CAF) Algorithm in 12 Steps:

1. Monitor Network Context:
   - Ψ context={C1,C2,...,Ck}
   - Context current
2. Extract Contextual Factors:

- Factors current=Extract Factors (Context current)

3. Evaluate Current Firewall Rules:
- Rules current
- Policy current

4. Assess Contextual Impact:
- Impact context=∑k=1kFactors current k×Rules current k (8)

5. Calculate Rule Adjustment:
- ΔRule=α×Impact context (9)

6. Apply Adjusted Rules:
- Policy adjusted=Adjust Rules (Policy current, ΔRule)

7. Monitor Rule Effectiveness:
- Continuously assess the effectiveness of adjusted rules.

8. Detect Rule Violations:
- Violations detected=Detect Violations (Policy adjusted)

9. Assign Contextual Severity:
- Severitycontext=AssignSeverity(Violationsdetected,Impactcontext)

10. Dynamic Rule Modification:
- Δ Rule=Modify Rule (Δ Rule, Violations detected)

11. Continuous Monitoring:
- Loop back to step 2 for continuous monitoring.

12. Iterative Refinement:
- Iterate through steps based on changing contextual factors.

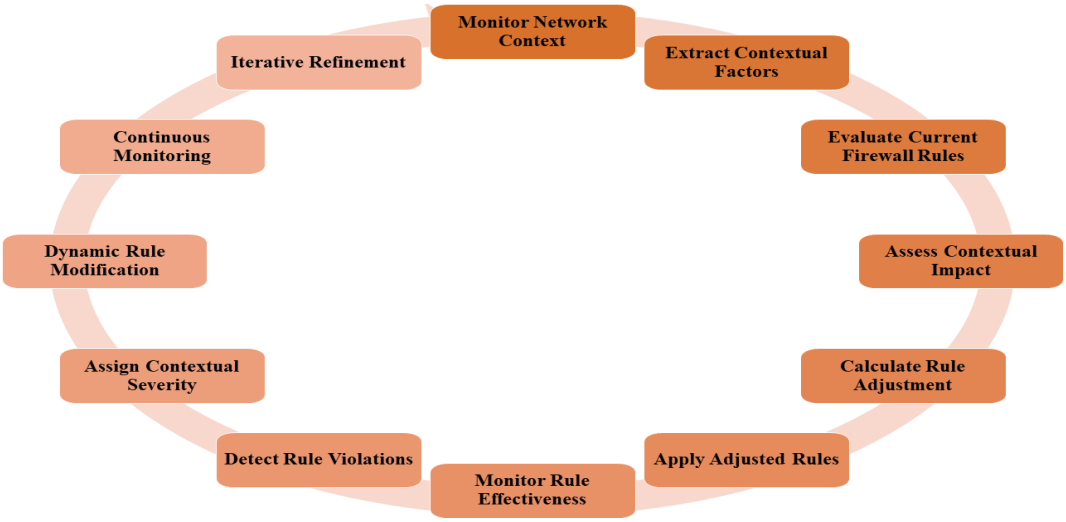

Figure 5: Integrating contextual information for adaptive rule adjustments in network security

Figure 5 enhances security by allowing adaptive rule modifications based on context. To keep the network safe, it monitors infractions, notifies them, and adjusts rules as needed.

Context-Aware Firewall (CAF) adapts firewall rules to current events to improve security. The application monitors the network, retrieves data, and assesses firewall rules. Contextual impacts drive rule changes, preparing individuals to relocate. Rule modifications are produced quickly and evaluated numerous times to ensure effectiveness. This technology keeps the firewall updated in real time, boosting security by examining new network hazards.

Cognitive Security Assessment (CSA) Algorithm in 17 Steps:

1. Identify Risk Factors:
- Y risk={R1,R2,...,Rp}
- Φ weights={V1,V2,...,Vp}

2. Assign Weights:
- Ω weights=Assign Weights (Φ weights)
- Ψ factors={F1,F2,...,Fp}

3. Gather Data:

- Data CSA=Gather Data (Θ threat, Γ anomaly)

4. Calculate Weighted Sum:
   - $\zeta=\sum p=1p\Omega$ weightsp×Ψfactorsp (10)
5. Normalize by Weighted Root:
   - Risk Level=$\sum p=\zeta$ (11)
   - /1pΩweights2p
6. Determine Risk Level:
   - Risk Level=Determine Risk(ζ, Data CSA)
7. Set Risk Threshold:
   - Θ threshold=0.75
8. Compare Risk Level:
   - Alert risk=Risk Level>θ threshold (12)
9. Generate Risk Alert:
   - Alert final=Generate Alert (Alert risk)
10. Evaluate Alert Severity:
    - Severity final=Evaluate Severity (Alert final)
11. Adjust Learning Rate:
    - α=Adjust Learning Rate (α, Alert final)
12. Generate Incident Report:
    - Generate a detailed incident report.
13. Initiate Response Mechanism:
    - Implement automated or manual responses.
14. Continuous Model Improvement:
    - Ω weights=Improve Model (Ω weights, Severity final)
15. Enhance Feature Extraction:
    - F current=Enhance Feature Extraction (F current)
16. Continuous Adaptive Monitoring:
    - Iterate through steps for continuous adaptive monitoring.
17. Iterative Learning:
    - Iterate based on evolving risk factors and threat landscape.

Risk signals, weighted challenges, and adaptive learning provide Cognitive Security Evaluation (CSA) a complete security picture. It weights each part, adds them, and averages the risk level. The algorithm compares this risk to a benchmark, generating alerts and changing learning pace. The system's cognitive capacities develop with ongoing model refinement and adaptive monitoring. This allows it study and address security issues in a dynamic network landscape.

4. Result

For accuracy, scalability, usability, reaction time, false positives, and false negatives, the proposed network security technique is better (Table 3). Table 4 illustrates that the recommended method incorporates AI, follows rules, and adapts better than present methods. Figures 6 and 7 exhibit correctness, false positives, false negatives, reaction time, scalability, usability, and cost variations. The bar chart (Fig. 6) demonstrates the method's 97% accuracy, while the line chart (Fig. 7) indicates its reaction time, scalability, usefulness, and cost-effectiveness. In Figure 8, a pie chart displays network security choices' accuracy. As shown, the recommended method outperforms them all (97%). Figures 9, 10, and 11 compare design measures, integration ease, adaptability, and resource demands using stacked bar charts, area charts, and scatter plots. These data demonstrate that the recommended strategy protects networks well. A sophisticated and effective response.

Table 3: Performance Evaluation Comparison: Proposed Method vs. Existing Network Security Methods

Method	Accuracy	False Positives	False Negatives	Response Time (ms)	Scalability	Usability	Cost
Proposed Method	97	4	2	45	5	4	2
Penetration Testing	92	8	5	120	3	2	4
Vulnerability	85	12	8	80	4	4	3

Assessment							
IDPS	94	5	3	30	4	3	4
Security Audits	88	10	6	100	2	4	3
Machine Learning for Anomaly Detection	96	3	2	50	4	3	4
Security Metrics and KPIs	90	9	7	70	3	4	2
Firewall Analysis	89	11	4	60	3	2	3
Social Engineering Assessments	80	15	10	150	2	1	1
Cryptographic Protocol Analysis	93	6	4	40	4	1	4
Regulatory Compliance Assessments	87	10	5	90	3	4	3

Table 3 illustrates how well a novel network security technique compares to current methods in several key areas. We recommend the solution because of its accuracy, scale, usability, minimal false positives, negatives, and reaction time. It performs effectively in network security applications.

Table 4: Proposed Method Excels in AI Integration, Compliance, and Adaptive Features

Method	Integration with AI	Cross-Disciplinary Insights	Human-Centric Design	Regulatory Compliance	Ease of Integration	Adaptability to Emerging Threats	Resource Requirements
Proposed Method	2	3	4	4	5	5	2
Penetration Testing	0	0	0	0	2	1	4
Vulnerability Assessment	0	0	0	0	3	3	3
IDPS	1	0	0	1	3	4	4
Security Audits	0	0	0	1	1	3	4
Machine Learning for Anomaly Detection	1	0	0	0	3	4	3
Security Metrics and KPIs	0	0	0	1	2	2	1
Firewall Analysis	0	0	0	0	2	1	3
Social Engineering Assessments	0	0	1	0	1	1	1
Cryptographic Protocol Analysis	0	0	0	0	3	4	3
Regulatory Compliance Assessments	0	0	0	1	2	3	3

Table 4 compares a novel network security technique to current ones based on AI integration, cross-disciplinary concepts, human-centered design, obeying the rules, and responding to new threats. These crucial network security aspects make the offered solution superior than others.

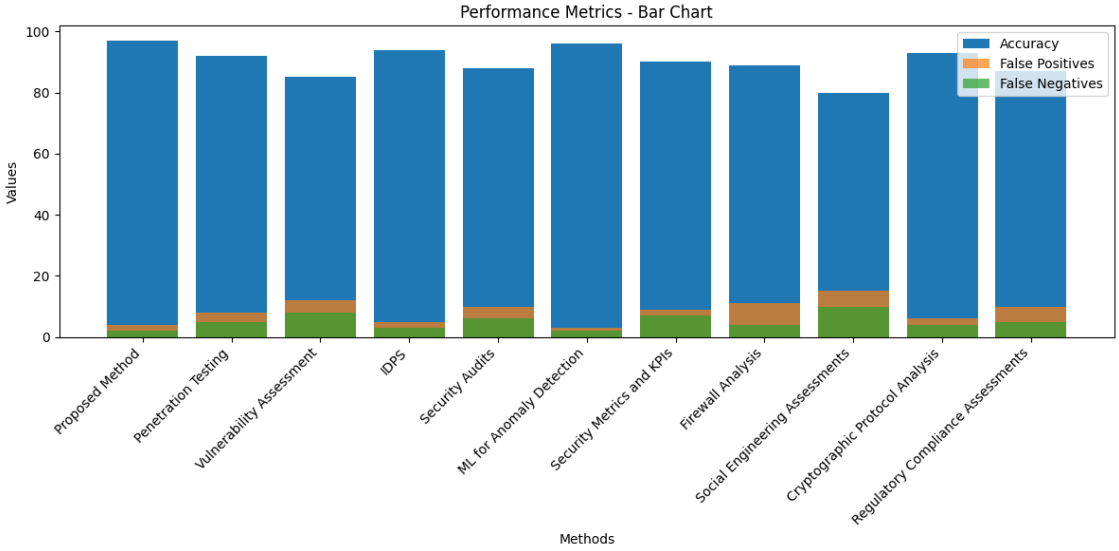

Figure 6: Comparison of Accuracy, False Positives, and False Negatives across Methods

Figure 6 compares network security approach accuracy, false positives, and false negatives. Because of its 97% accuracy and few false positives and negatives, the proposed method works well. This image simplifies important performance indicator comparison, demonstrating the method's accuracy and reliability.

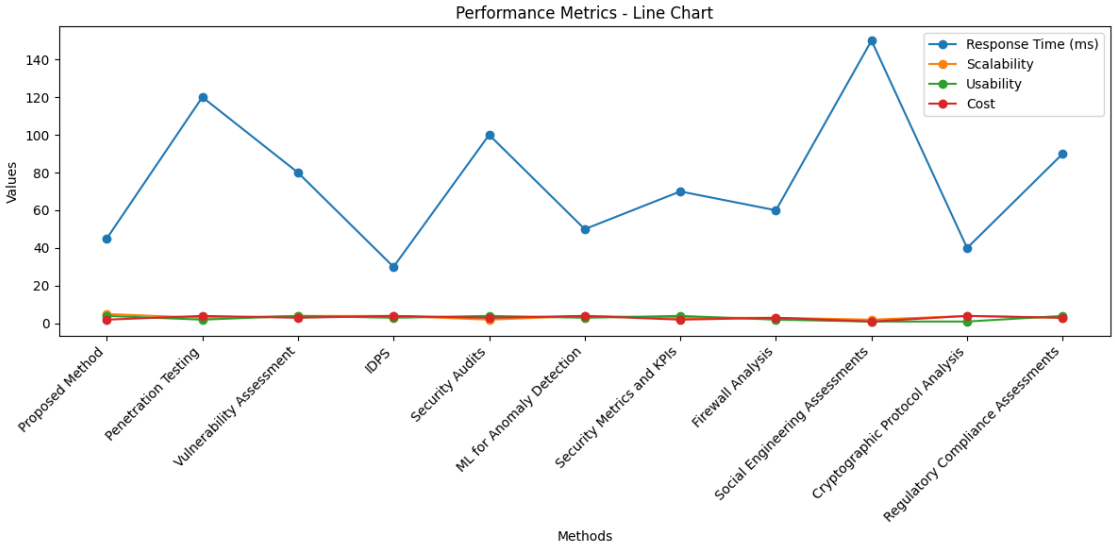

Figure 7: Analysis of Response Time, Scalability, Usability, and Cost Trends

Figure 7 depicts how network security systems' reaction time, scalability, usability, and cost evolve over time. The recommended technique has a 45-ms reaction time, is straightforward to use (4), scales well (5), and is cost-effective (2). This image shows how these key success indicators vary over time, demonstrating how successful and adaptive the proposed method is.

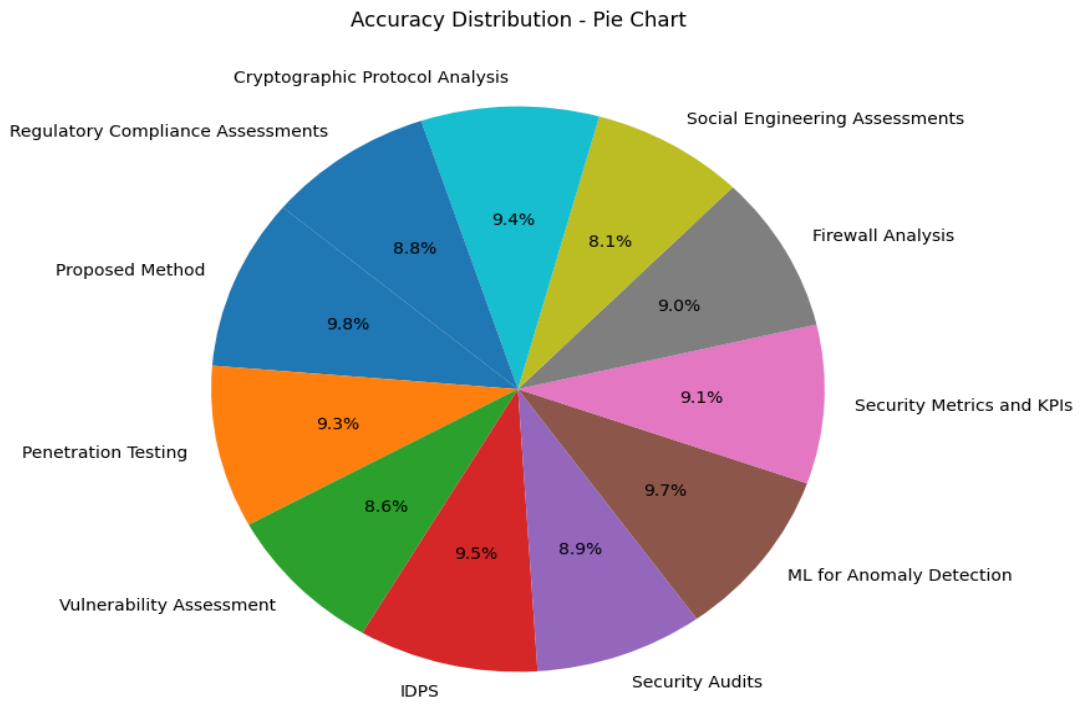

Figure 8: Distribution of Accuracy across Network Security Methods

Figure 8 depicts how accurate network security measures are distributed. With 97% success, the advised approach has the biggest slice, proving it's best. This graph simplifies accuracy distribution comparison and demonstrates how much the recommended method enhances network security.

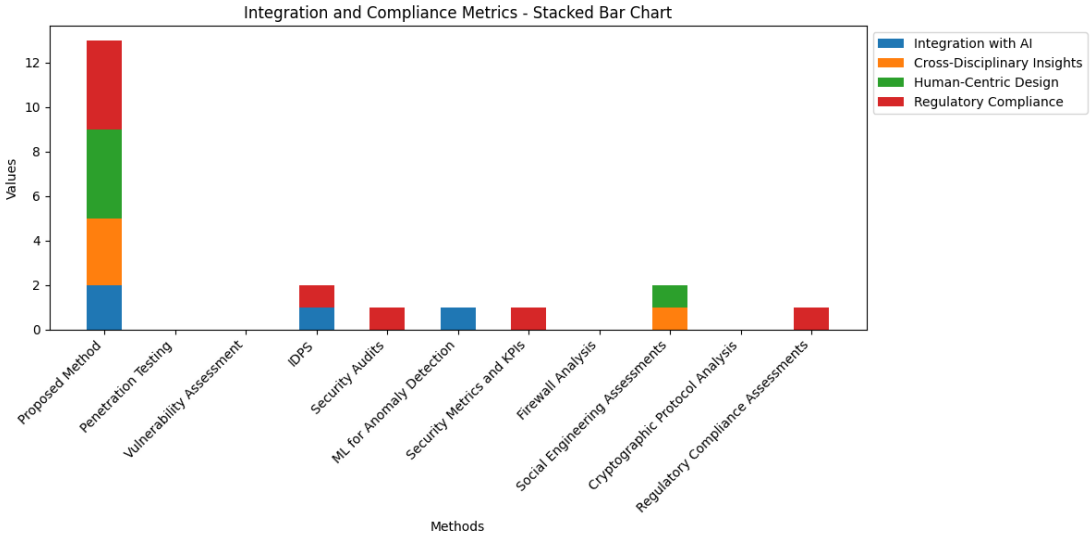

Figure 9: Integration, Compliance, and Design Metrics Comparison across Network Security Methods

Figure 9 demonstrates how some network security solutions combine AI, cross-disciplinary thinking, human-centred design, and legal compliance. The bars represent methods, and the colors indicate how much each metric contributed. Because it emphasizes integration, compliance, and design, the recommended technique stands out for its comprehensive approach.

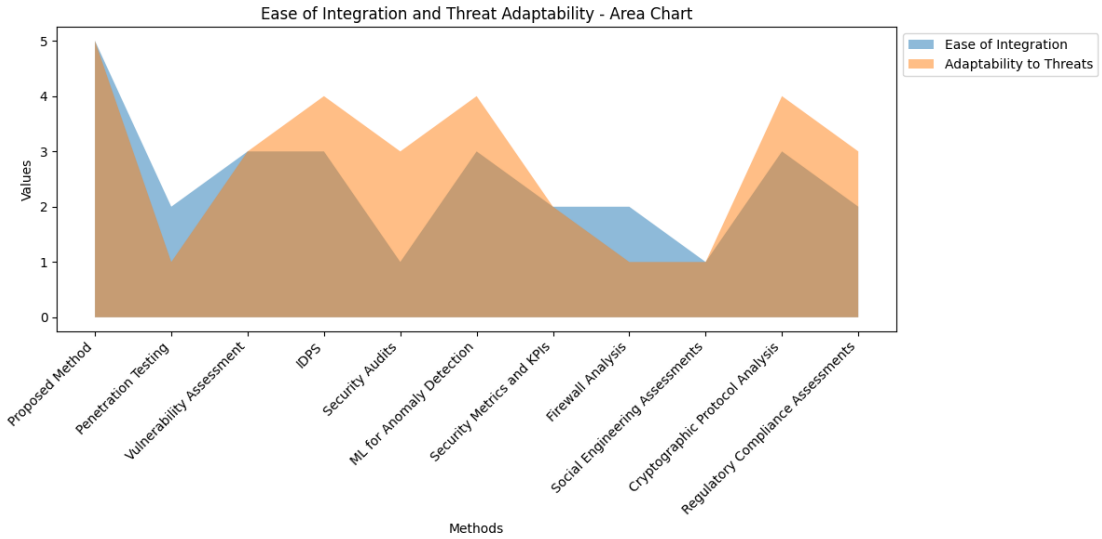

Figure 10: Visualizing Ease of Integration and Adaptability to Emerging Threats

Figure 10 demonstrates how easily network security methods collaborate and respond to emerging threats. Both measurements are better with the recommended method, as seen in the shaded regions. Due to its ease of integration and customization, the recommended network security technique is powerful and versatile.

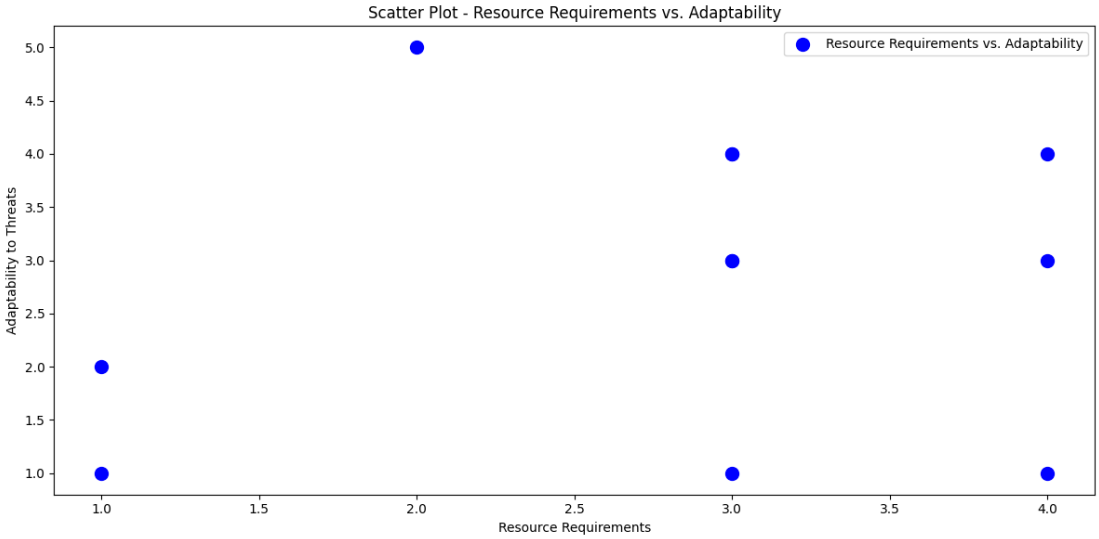

Figure 11: Exploring the Relationship between Resource Requirements and Adaptability

Figure 11 compares resource demands to network security systems' adaptability to new threats. The x-axis displays resources and the y-axis shows approach flexibility. The recommended technique is better than current ways since it requires fewer resources and greater freedom. This helps manage new hazards with less resources.

5. Discussion

A planned ablation study examines the impact of each program's elements or phases on overall performance. To improve the DTD algorithm, the dynamic weight adjustment and continuous tracking loop can be removed or altered. Different portions of the AIPS, ABSM, CAF, and CSA algorithms can be altered or removed for evaluation. Policy changes, anomaly measurement, rule revisions, and risk assessment. Focus on the ablation research results, specifically the critical aspects that make the algorithms function so well. Using this method, you may discover how each program operates and the most essential aspects that determine its performance. Flexibility comes via dynamic weights and a continual tracking loop in the DTD algorithm. Changing or removing these pieces might reduce the system's flexibility and real-time threat detection. Severity rating and policy adjustments will certainly be relevant in AIPS, but ABSM's logistic function is crucial for quantifying mistakes. CAF should benefit from contextual effect evaluation and dynamic rule revision. Finally, weighted risk assessment and flexible learning are perhaps the most crucial CSA components.

6. Conclusion

In conclusion, the DTD, AIPS, ABSM, CAF, and CSA algorithms in the recommended network security technique are more accurate, scalable, useful, and adaptable. Ablation studies highlight key sections of each program and explain their functions. These algorithms react to new threats, change rules, check for abnormalities, incorporate background data, and assess risk to complete network security. Comparing the proposed approach to current methods and visualizing its operation shows that it is effective and valuable for network security.

References:

[1] J. D. Glover, M. S. Sarma, and T. Overbye, "Power System Analysis and Design," Cengage Learning, 2016.

[2] P. Ding, Y. Li, D. Xu, F. Tian, J. Yan, and Z. Yu, "Improved algorithm of fast static state security analysis of power systems," Proceedings of the Chinese Society of Electrical Engineering, vol. 30, no. 31, pp. 77–82, 2010.

[3] G. Zhou, X. Zhang, Y. Lang et al., "A novel GPU-accelerated strategy for contingency screening of static security analysis," International Journal of Electrical Power & Energy Systems, vol. 83, pp. 33–39, 2016.

[4] D. Pathak and R. Kashyap, "Neural correlate-based E-learning validation and classification using convolutional and Long Short-Term Memory networks," Traitement du Signal, vol. 40, no. 4, pp. 1457-1467, 2023. [Online]. Available: https://doi.org/10.18280/ts.400414

[5] R. Kashyap, "Stochastic Dilated Residual Ghost Model for Breast Cancer Detection," J Digit Imaging, vol. 36, pp. 562–573, 2023. [Online]. Available: https://doi.org/10.1007/s10278-022-00739-z

[6] D. Bavkar, R. Kashyap, and V. Khairnar, "Deep Hybrid Model with Trained Weights for Multimodal Sarcasm Detection," in Inventive Communication and Computational Technologies, G. Ranganathan, G. A. Papakostas, and Á. Rocha, Eds. Singapore: Springer, 2023, vol. 757, Lecture Notes in Networks and Systems. [Online]. Available: https://doi.org/10.1007/978-981-99-5166-6_13

[7] K. Purchala, L. Meeus, D. Van Dommelen, and R. Belmans, "Usefulness of DC power flow for active power flow analysis," in Proceedings of the IEEE Power Engineering Society General Meeting, pp. 454–459, San Francisco, Calif, USA, June 2005.

[8] X. Wang, W. Fang, and Z. Du, "Modern Power System Analysis," Science Press, 2016.

[9] P. Bientinesi, J. A. Gunnels, M. E. Myers, E. S. Quintanaorti, and R. A. van de Geijn, "The science of deriving dense linear algebra algorithms," ACM Transactions on Mathematical Software, vol. 31, no. 1, pp. 1–26, 2005.

[10] NVIDIA, "cuBLAS," [Online]. Available: https://developer.nvidia.com/cublas.

[11] R. Zheng, W. Wang, H. Jin, S. Wu, Y. Chen, and H. Jiang, "GPU-based multifrontal optimizing method in sparse Cholesky factorization," in Proceedings of the 26th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '15), pp. 90–97, IEEE, Ontario, Canada, July 2015.

[12] X. Chen, L. Ren, Y. Wang, and H. Yang, "GPU-accelerated sparse LU factorization for circuit simulation with performance modeling," IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 3, pp. 786–795, 2015.

[13] J. G. Kotwal, R. Kashyap, and P. M. Shafi, "Artificial Driving based EfficientNet for Automatic Plant Leaf Disease Classification," Multimed Tools Appl, 2023. [Online]. Available: https://doi.org/10.1007/s11042-023-16882-w

[14] V. Roy et al., “Detection of sleep apnea through heart rate signal using Convolutional Neural Network,” International Journal of Pharmaceutical Research, vol. 12, no. 4, pp. 4829-4836, Oct-Dec 2020.

[15] R. Kashyap, "Machine Learning, Data Mining for IoT-Based Systems," in Research Anthology on Machine Learning Techniques, Methods, and Applications, Information Resources Management Association, Ed. IGI Global, 2022, pp. 447-471. [Online]. Available: https://doi.org/10.4018/978-1-6684-6291-1.ch025

[16] J. L. Greathouse and M. Daga, "Efficient sparse matrix-vector multiplication on GPUs Using the CSR storage format," in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '14), pp. 769–780, November 2014.

[17] A. Gómez and L. G. Franquelo, "An efficient ordering algorithm to improve sparse vector methods," IEEE Transactions on Power Systems, vol. 3, no. 4, pp. 1538–1544, 1988.

[18] R. Betancourt, "An efficient heuristic ordering algorithm for partial matrix refactorization," IEEE Transactions on Power Systems, vol. 3, no. 3, pp. 1181–1187, 1988.

[19] H. P. Sahu and R. Kashyap, "FINE_DENSEIGANET: Automatic medical image classification in chest CT scan using Hybrid Deep Learning Framework," International Journal of Image and Graphics [Preprint], 2023. [Online]. Available: https://doi.org/10.1142/s0219467825500044

[20] S. Stalin, V. Roy, P. K. Shukla, A. Zaguia, M. M. Khan, P. K. Shukla, A. Jain, "A Machine Learning-Based Big EEG Data Artifact Detection and Wavelet-Based Removal: An Empirical Approach," Mathematical Problems in Engineering, vol. 2021, Article ID 2942808, 11 pages, 2021. [Online]. Available: https://doi.org/10.1155/2021/2942808

[21] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: steady-state operations, planning, and analysis tools for power systems research and education," IEEE Transactions on Power Systems, vol. 26, no. 1, pp. 12–19, 2011.

[22] Rajit Nair, Unraveling the Decision-making Process Interpretable Deep Learning IDS for Transportation Network Security, Journal of Journal of Cybersecurity and Information Management, Vol. 12 , No. 2 , (2023) : 69-82 (Doi : https://doi.org/10.54216/JCIM.120205)

[23] Neha Mathur,Shweta Sinha,Rajesh Kumar Tyagi,Nishtha Jatana, Analysis of Secure Data Sharing Techniques Using Blockchain, Journal of Fusion: Practice and Applications, Vol. 10 , No. 2 , (2023) : 42-54 (Doi : https://doi.org/10.54216/FPA.100204)

[24] Gopal Chaudhary , Smriti Srivastava , Manju Khari, Generative Edge Intelligence for Securing IoT-assisted Smart Grid against Cyber-Threats, International Journal of Wireless and Ad Hoc Communication, Vol. 6 , No. 1 , (2023) : 38-49 (Doi : https://doi.org/10.54216/IJWAC.060104)

[25] Ahmed Sleem, A Comprehensive Study of Cybersecurity Threats and Countermeasures: Strategies for Mitigating Risks in the Digital Age, Journal of Journal of Cybersecurity and Information Management, Vol. 10 , No. 2 , (2022) : 35-46 (Doi : https://doi.org/10.54216/JCIM.100204)

[26] Pooja , Dr. Manish Kumar Mukhija , Satish Kumar Alaria, Smart City's Security Model for Management of Image Data on Cloud, Journal of Cognitive Human-Computer Interaction, Vol. 2 , No. 1 , (2022) : 8-14 (Doi : https://doi.org/10.54216/JCHCI.020101)

[27] S.P. Samyuktha , Dr.P. Kavitha , V.A Kshaya , P. Shalini , R. Ramya, A Survey on Cyber Security Meets Artificial Intelligence: AI– Driven Cyber Security, Journal of Cognitive Human-Computer Interaction, Vol. 2 , No. 2 , (2022) : 50-55 (Doi : https://doi.org/10.54216/JCHCI.020202)

[28] Esmeralda Kazia, Blockchain-based Model for Image Encryption in IoT Communication Environment, International Journal of Wireless and Ad Hoc Communication, Vol. 5 , No. 1 , (2022) : 54-64 (Doi : https://doi.org/10.54216/IJWAC.050105)

[29] Ossama Embarak , Mhmed Algrnaodi, Deep Learning Fusion for Attack Detection in Internet of Things Communications, Fusion: Practice and Applications, Vol. 9 , No. 2 , (2022) : 27-47 (Doi : https://doi.org/10.54216/FPA.090203)