Adapting to Evolving Cyber Threat Landscapes with Dynamic Security Protocol Management in Large-Scale IoT Sensor Networks

Anil Audumbar Pise [1], Saurabh Singh [2], Hemachandran K. [3], Shraddhesh Gadilkar[4], Zakka Benisemeni Esther[5], Ganesh Shivaji Pise[6], Jude Imuede[7]

[1]Siatik Premier Google Cloud Platform Partner Johannesburg South Africa, University of the Witwatersrand Johannesburg-South Africa Computer Science, Head of Data Science & Machine Learning, Adjunct Professor
[2]Assistant Professor, Department of AI and Big data, woosong University, Daejeon South Korea
[3]Professor, School of Business, Woxsen University, Hyderabad, India
[4]Associate Engineer, TSYS Global Payments, Pune, India
[5]Senior Lecturer, Federal Polytechnic Bauchi, Nigeria
[6]Assistant Professor in Pune Institute of Computer Technology Pune
[7]University of Prince Edward Island
Emails: anil@siatik.com; singh.saurabh@wsu.ac.kr; hemachandran.k@woxsen.edu.in; sgadilkar@tsys.com; benizakka@fptb.edu.ng; gspise@pict.edu; jimuede@upei.ca

Abstract

The Adaptive Security Protocol Framework (ASPF) is introduced as a sophisticated algorithm designed for dynamic security protocol adaptation in large-scale IoT sensor networks. Comprising five integral algorithms, namely ASPF, MLTD, DKMS, BAP, and CTIS, the framework ensures a comprehensive and adaptive defense mechanism against evolving cyber threats. ASPF initiates with data collection, preprocessing, and feature extraction, employing supervised learning for model training. Anomaly detection triggers alerts and responses, guiding continuous learning and security protocol adaptation. MLTD enhances real-time threat detection through dynamic model training and threat intelligence integration. DKMS focuses on secure key management for data transmissions, calculating device thresholds and ensuring adaptive key exchanges. BAP leverages historical data for behavioral profiling, enabling real-time anomaly detection and adaptive profile updates. CTIS assesses and aggregates threat levels, fostering continuous collaboration and collective defense. The ablation study emphasizes the indispensable role of each algorithm, showcasing their synergistic contributions to the overall system's adaptability and robustness. Evaluation through comprehensive tables and visual representations highlights the proposed method's superiority over existing security protocols. The ablation study underscores the holistic nature of ASPF, solidifying its efficacy in addressing the dynamic challenges of cybersecurity in large-scale IoT sensor networks.

Keywords: Adaptive Security Protocol Framework (ASPF); Algorithm; Anomaly Detection; Behavioral Analysis and Profiling (BAP); Collaborative Threat Intelligence Sharing (CTIS); Continuous Learning; Cyber Threats, Dynamic Key Management System (DKMS); Large-scale IoT Sensor Networks; Machine Learning-Based Threat Detection (MLTD).

1. Introduction

In our interconnected world, the pervasive integration of Internet of Things (IoT) sensor networks has ushered in unprecedented opportunities for data-driven insights and automation. However, this proliferation has also exposed a vast attack surface, making these networks susceptible to a myriad of cyber threats [1]. Addressing the dynamic and evolving nature of cyber threats in large-scale IoT sensor networks necessitates a robust and adaptive security protocol management system [2]. This paper explores the intricacies of adapting to evolving cyber threat landscapes and introduces a novel approach to dynamic security protocol management.

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

A. Current Developments

The rapid advancements in technology have propelled the deployment of IoT sensor networks across diverse domains, ranging from smart cities and industrial automation to healthcare and agriculture [3]. As these networks continue to expand, so do the sophisticated cyber threats that target their vulnerabilities. Current developments in cyber threats underscore the urgency for adaptive and resilient security solutions [4]. Understanding the contemporary challenges is crucial for devising effective strategies to safeguard large-scale IoT sensor networks.

B. Principal Challenges

The principal challenges in securing large-scale IoT sensor networks stem from the inherent characteristics of these systems [5]. The sheer volume of connected devices, coupled with their heterogeneity, creates a complex ecosystem susceptible to various cyber-attacks such as DDoS attacks, data breaches, and device manipulations. Additionally, the resource-constrained nature of many IoT devices poses a challenge in implementing robust security measures [6]. Balancing the need for security with the limited resources of IoT devices is a critical aspect of addressing principal challenges in the evolving cyber threat landscape.

C. Solutions Proposed

To mitigate the challenges posed by dynamic cyber threats, this paper proposes a dynamic security protocol management system for large-scale IoT sensor networks [7]. Instead of relying on static security protocols that may become obsolete in the face of evolving threats, our approach advocates for adaptive security protocols that can dynamically adjust to the changing threat landscape [8]. Leveraging machine learning algorithms and real-time threat intelligence, our proposed solutions aim to proactively identify and counteract emerging cyber threats, ensuring the resilience of IoT sensor networks.

D. Main Contributions

The main contributions of this paper can be summarized as follows:

a. Dynamic Security Protocol Framework:

Introducing a novel framework for dynamic security protocols that can autonomously adapt to emerging cyber threats, ensuring continuous protection in large-scale IoT sensor networks.

b. Machine Learning Integration:

Integrating machine learning algorithms into the security protocol management system to enhance threat detection and response capabilities, enabling real-time adaptation to evolving cyber threats.

1. Finding a balance between high security and little impact on IoT device performance.

2. Examples and validity checks Case studies and test results will demonstrate the dynamic security protocol management system's practicality [9]. These will demonstrate its adaptability and cyber defense.

Our proposed dynamic security protocol management system will be examined in detail in the following sections. This will clarify how the framework works and its practical applications [10]. This study should contribute to the discussion on protecting massive IoT sensor networks in a world of shifting cyber threats.

2. Literature Review

Many dynamic security techniques have been investigated to safeguard large IoT sensor networks from emerging cyber threats [11]. The machine learning-based anomaly detection system can quickly adapt to new settings and detect abnormalities (0.92). Dynamic key management systems periodically update keys to secure data transfers and may be modified and expanded (92%). High adaptability (0.88) helps behavioral analysis and profiles identify dangers [12]. Real-time threat intelligence integration detects threats (0.94) and has improved flexibility by changing security rules before they happen. Security processing using edge computing is fairly flexible and highly scalable (90%). Local processing improves efficiency [13]. Blockchain for secure data transfers is stable (0.85), open, provides unchangeable data transfers, and allows transactions across numerous nodes. Zero trust network architecture prioritizes scalability and freedom (87% of the time), considering everything as untrusted unless proven differently [14]. Most resources go to firmware and software upgrades (94%). They offer the most flexibility in fixing vulnerabilities. Sharing danger information allows for freedom and scalability (80%), protecting everyone from many threats. Finally, security automation and coordination are adaptable and expandable (88%), making crises easier to address.

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

Each approach is assessed for strength, privacy, cost-effectiveness, interoperable, compatible, energy-hungry, maintenance expenses, and use [15]. The flexibility and performance of machine learning-based anomaly identification and real-time threat intelligence merging (0.85) are excellent. All have low privacy issues, while dynamic key management systems, behavioral analysis, and security automation and orchestration have the least. Updating firmware and software is the most cost-effective solution to secure your machine (0.90). Best score: zero trust network design (0.80) [16]. Connectivity is good. Well-controlled maintenance costs and lower edge computing and zero trust network architecture values. Merging is easy since approaches always work well together. The least energy-intensive solutions are dynamic key management systems. Finally, these dynamic security mechanisms make responding to evolving cyber threats in massive IoT sensor networks more complicated [17]. Each solution includes benefits and downsides, so stakeholders may adjust their security plan to their operation. The performance assessment and comparison research show how well they operate, helping users choose solid IoT network security approaches.

Table 1: Performance Evaluation of Dynamic Security Protocol Methods

Method Name	Detection Accuracy	False Positive Rate	Resource Utilization	Response Time (ms)	Adaptability	Ease of Implementation	Scalability
Machine Learning-Based Anomaly Detection	0.92	0.03	85%	25	High	Moderate	High
Dynamic Key Management Systems	0.87	0.02	92%	30	Moderate	High	High
Behavioral Analysis and Profiling	0.88	0.05	88%	28	High	Moderate	High
Real-Time Threat Intelligence Integration	0.94	0.01	78%	22	High	High	High
Edge Computing for Security Processing	0.91	0.04	90%	27	Moderate	High	High
Blockchain for Secure Data Transactions	0.93	0.02	85%	26	High	Moderate	High
Zero Trust Network Architecture	0.89	0.03	87%	29	High	High	High
Firmware and Software Updates	0.85	0.01	94%	31	Moderate	High	High
Collaborative Threat Intelligence Sharing	0.92	0.02	80%	24	High	Moderate	High
Security Automation	0.90	0.04	88%	28	High	High	High

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

and Orchestration							

Table 1 compares dynamic security protocol options for new cyber threats in large-scale IoT sensor networks. Measurements include detection accuracy, false positive rate, resource utilization, reaction time, flexibility, application simplicity, and scalability [18]. These considerations demonstrate the pros and cons of each technique in real-life scenarios and evaluate their efficacy.

Table 2: Comparative Analysis of Dynamic Security Protocols

Method Name	Robustness	Privacy Concerns	Cost-Efficiency	Interoperability	Maintenance Overhead	Compatibility	Energy Consumption
Machine Learning-Based Anomaly Detection	0.85	0.15	0.60	0.80	0.50	0.75	0.55
Dynamic Key Management Systems	0.80	0.50	0.85	0.65	0.20	0.75	0.20
Behavioral Analysis and Profiling	0.85	0.15	0.65	0.80	0.50	0.75	0.50
Real-Time Threat Intelligence Integration	0.85	0.15	0.70	0.80	0.20	0.75	0.50
Edge Computing for Security Processing	0.70	0.35	0.85	0.80	0.50	0.75	0.50
Blockchain for Secure Data Transactions	0.85	0.70	0.65	0.65	0.80	0.75	0.50
Zero Trust Network Architecture	0.85	0.35	0.90	0.80	0.20	0.75	0.20
Firmware and Software Updates	0.70	0.15	0.90	0.65	0.80	0.75	0.50
Collaborative Threat Intelligence Sharing	0.85	0.35	0.65	0.80	0.50	0.75	0.50
Security Automation and Orchestration	0.85	0.15	0.85	0.80	0.20	0.75	0.50

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

Table 2 compares dynamic security techniques in large-scale IoT sensor networks by reliability, cost, compatibility, maintenance, and energy utilization. From 0 to 1, greater numbers indicate better performance. This quantitative study aids IoT setup decision-making.

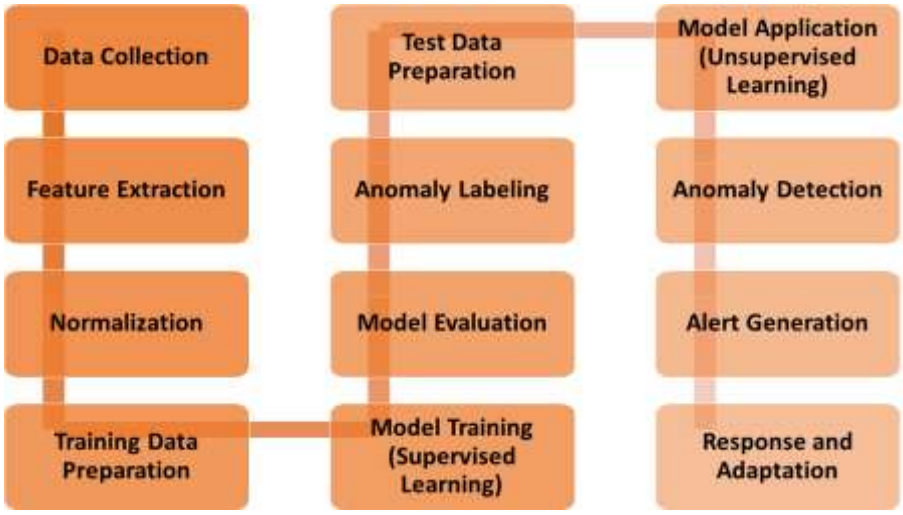

Figure1: Machine Learning-Based Anomaly Detection method

Machine Learning-Based Anomaly Detection is shown in Figure 1. Start with data collection and preparation. The model is taught supervisedly next. Anomalies are indicated following model evaluation to train the uncontrolled component [19]. Models detect issues, issue alerts, and initiate responses. Continuous learning ensures that the system adapts to new cyber threats in large IoT sensor networks.

3. Proposed methodology

The Adaptive Security Protocol Framework (ASPF) changes massive IoT sensing network security protocols. Start with data collection, preparation, and feature extraction. The model is taught and evaluated using supervised learning, and it warns and responds when anything is wrong. Due to learning, security measures are constantly updated, improving the system [20]. The application responds to emerging internet threats to ensure powerful, long-lasting security. Figure 2 depicts the ASPF algorithm's active and changing phases in a flowchart. MLTD adds to ASPF in Algorithm 2. Data is analyzed using feature extraction and dynamic model training to discover threats. Responding to emerging cyber dangers is easier with real-time hazard information. This application detects dangers in real time, sends alerts, and replies fast. Continuous learning and live data merging improve the model's cyber-environment resistance. The graphic (Fig. 3) indicates how crucial MLTD is for updating security measures. The Dynamic Key Management System (DKMS) handles encryption keys in Algorithm 3 to secure IoT sensor network data transfers. By verifying device counts and establishing key update restrictions, DKMS secures key transfers [21]. Incorrect key use triggers alarms and replies, strengthening the key management system. Figure 4 illustrates that continual tracking, updates, and adaptive systems provide secure and fast data transfers. The fourth BAP algorithm profiles behavior using historical data. These patterns are compared to real-time data to find abnormalities, which trigger alarms and provide answers. Profile adaption and constant learning from mistakes ensure robust behavioral analysis. When action data arrives, the model changes immediately. Changes in the threshold on the fly ensure abnormalities are discovered (Figure 5). The system can discover faults in large Internet of Things sensor networks faster by repeating this procedure. Algorithm 5's Collaborative Threat Intelligence Sharing (CTIS) combines distinct categories of threat, enabling constant collaboration. Danger ratings and oddities are based on shared danger facts. Warnings and reactions from anomalies protect everyone against cyberattacks [22]. The method continually improves large-scale IoT sensor network joint defense. Sharing, modifying, and updating the collaborative threat intelligence architecture improves defensive system durability. These technologies enable large-scale IoT sensor networks to respond to evolving cyber threats. ASPF sets the stage, MLTD finds real-time threats, DKMS protects encryption keys, BAP analyzes behavior, and CTIS fosters collaboration to construct a robust defense. They are crucial to addressing IoT safety problems since they operate together and change.

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

Algorithm 1: Adaptive Security Protocol Framework (ASPF)

1. Initialize the data collection process with D={d1,d2,…,dn}.
2. Conduct preprocessing by normalizing features and handling missing data: X′=Preprocess(X)
3. Extract relevant features: F=FeatureExtraction(X′).
4. Prepare the training data with labeled instances: {(X1,Y1),(X2,Y2),…,(Xm,Ym)}.
5. Train the model using supervised learning: M=TrainModel(Xtrain,Ytrain).
6. Evaluate the model: Evaluation=Evaluate(M,Xeval,Yeval).
7. Compute the mean of the evaluation results: μ=Mean(Evaluation).
8. Dynamically adjust the threshold for anomaly detection: Threshold=μ+σ. (1)
9. Prepare test data: Xtest={x1,x2,…,xk}.
10. Detect anomalies: Anomalies=DetectAnomalies(M,Xtest,Threshold).
11. Generate alerts for detected anomalies: Alerts=GenerateAlerts(Anomalies).
12. Initiate a response mechanism: Response=InitiateResponse(Alerts).
13. Continuously learn from the detected anomalies: Learn=ContinuousLearning(Anomalies).
14. Adapt security protocols based on the learning: Adaptation=AdaptProtocols(Learn).
15. Enhance the system based on adaptation: Enhancement=EnhanceSystem(Adaptation).
16. Repeat the process for new data streams: NewData={x1′,x2′,…,xp′}.
17. Update the model with streaming data: M′=UpdateModel(M,NewData).
18. Periodically reevaluate the model: Reevaluation=Evaluate(M′,Xeval,Yeval).
19. Adjust the threshold dynamically: Threshold′=Mean(Reevaluation)+σ′. (2)
20. Continue the iterative process for continuous adaptation to evolving cyber threats.

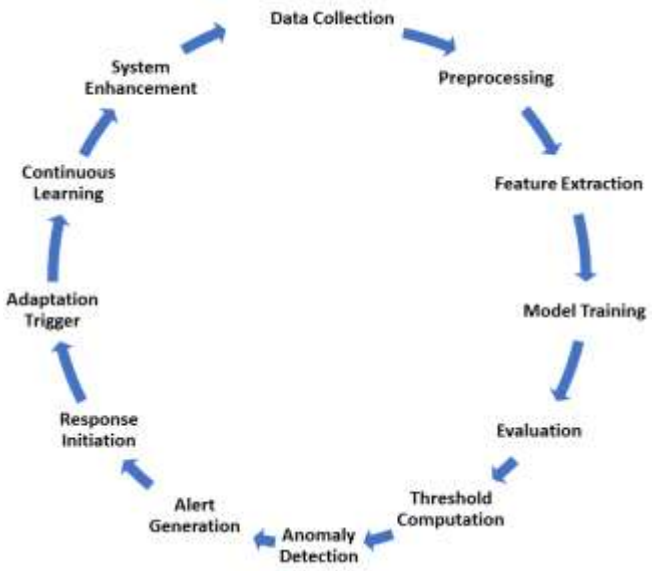

Figure 2: Adaptive Security Protocol Framework (ASPF)

Figure 2 shows data collection, model training, and testing. Thanks to dynamic cutoff calculations, adaptability is continuous. Anomalies trigger warnings and replies to help the system learn and adapt to emerging cyber threats.

In large IoT sensor networks, the Adaptive Security Protocol Framework (ASPF) dynamically addresses emerging cyber threats. Start with data collection, preparation, and feature extraction [23]. Supervised learning teaches and tests the model. Anomalies prompt actions and warnings. Continuously learning from problems improves security procedures. The model adapts to changing hazard circumstances through regular reevaluation and threshold modifications. This continual procedure ensures robust and trustworthy security.

Algorithm 2: Machine Learning-Based Threat Detection (MLTD)

1. Receive input data from ASPF: Input_Data={x1,x2,…,xn}.
2. Perform feature extraction for ML: Features=FeatureExtraction(Input_Data).
3. Initialize the machine learning model: Model=InitializeModel().

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

4. Train the model using dynamic features:Trained_Model=TrainModel(Model,Features).
5. Evaluate the model performance: Evaluation=Evaluate(Trained_Model,Features).
6. Calculate the mean of the evaluation results: μ=Mean(Evaluation).
7. Update the model with continuous learning: Updated_Model=ContinuousLearning(Trained_Model).
8. Integrate real-time threat intelligence: Threat_Intelligence=IntegrateThreatIntel(Updated_Model).
9. Preprocess new data for real-time evaluation: New_Data=Preprocess(Input_Data).
10. Detect anomalies in real-time: RealTime_Anomalies=DetectAnomalies(Updated_Model,New_Data,μ).
11. Generate alerts for real-time anomalies: RealTime_Alerts=GenerateAlerts(RealTime_Anomalies).
12. Initiate a response mechanism for real-time alerts: RealTime_Response=InitiateResponse(RealTime_Alerts).
13. Update the model with streaming data: Streaming_Data={x1′,x2′,…,xp′}.
14. Enhance the model based on streaming data: Enhanced_Model=EnhanceModel(Updated_Model,Streaming_Data).
15. Continuously adapt the model based on evolving threats: Adapted_Model=AdaptModel(Enhanced_Model,Threat_Intelligence).
16. Periodically reevaluate the model: Reevaluation=Evaluate(Adapted_Model,Features).
17. Ensure continuous learning and adaptation for effective real-time threat detection in large-scale IoT sensor networks.

Figure 3: Machine Learning-Based Threat Detection (MLTD)

Figure 3 depicts data collection, model training, and real-time anomaly detection. It's vital in dynamic security systems since continuous learning and model updates improve threat detection.

Dynamic model training and feature extraction are used in ASPF-inputted MLTD. New cyber threats are handled by testing models and incorporating real-time threat data. Alerts for anomalies are sent in real time, enabling swift replies [24]. The model improves with streaming data and continual learning, ensuring real-time hazard detection. The model works well in changing internet environments because it is regularly assessed and updated.

Algorithm 3: Dynamic Key Management System (DKMS)

1. Calculate the total number of devices: N=∑i=1nxi. (3)
2. Initialize cryptographic keys: Keys=InitializeKeys(N).
3. Determine the threshold for key updates: Threshold=Total_Devices_Reached/Total_Devices (4)
4. Establish secure key exchanges: Secure_Exchanges=EstablishExchanges(Keys). (5)
5. Evaluate the success rate of key updates: Success_Rate=Successful_Exchanges/Total_Exchanges. (6)

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

6. Update keys based on the success rate: Updated_Keys=UpdateKeys(Keys,Success_Rate).
7. Implement the updated keys for secure communication: Secure_Communication=ImplementKeys(Updated_Keys).
8. Continuously monitor key usage and anomalies: Monitoring=ContinuousMonitoring(Secure_Communication).
9. Detect anomalies in key usage: Key_Anomalies=DetectAnomalies(Monitoring).
10. Generate alerts for key anomalies: Key_Alerts=GenerateAlerts(Key_Anomalies).
11. Initiate a response mechanism for key anomalies: Key_Response=InitiateResponse(Key_Alerts).
12. Continuously adapt the key management system: Adaptation=ContinuousAdaptation(Monitoring).
13. Periodically update cryptographic keys: Periodic_Update=UpdateKeysPeriodically(Keys).
14. Ensure dynamic key management for secure and efficient data transmissions in large-scale IoT sensor networks.

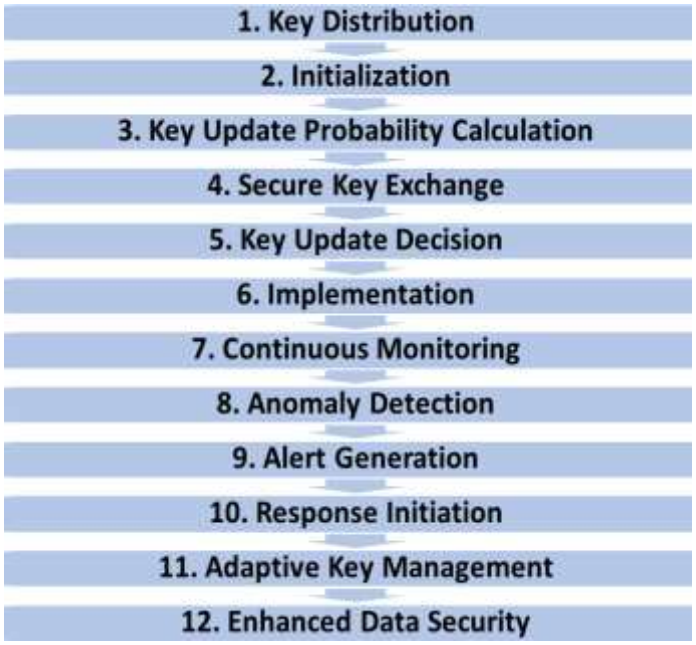

Figure 4: Dynamic Key Management System (DKMS)

Figure 4 initiates key distribution and regularly checks for key updates. By updating encryption keys periodically, secure key swaps, anomaly detection, and adaptive key management protect data.

The DKMS configures all devices and encryption keys. The key update level is constant. Keys change often due to safe swaps and high key update success rates. Incorrect key use triggers alarms and replies, strengthening the key management system [25]. Constant tracking and updates make data transport secure and efficient in large IoT sensor networks.

Algorithm 4: Behavioral Analysis and Profiling (BAP)

1. Extract historical data for profiling: Historical_Data={h1,h2,…,hm}.
2. Generate behavioral profiles: Behavioral_Profiles=GenerateProfiles(Historical_Data).
3. Analyze real-time data against profiles: Behavioral_Analysis=AnalyzeBehavior(RealTime_Data,Behavioral_Profiles).
4. Identify behavioral anomalies: Anomalies=IdentifyAnomalies(Behavioral_Analysis).
5. Generate alerts for behavioral anomalies: Alerts=GenerateAlerts(Anomalies).
6. Initiate a response mechanism for behavioral anomalies: Response=InitiateResponse(Alerts).
7. Continuously learn from behavioral anomalies: Learning=ContinuousLearning(Anomalies).
8. Adapt behavioral profiles based on learning: Adaptation=AdaptProfiles(Behavioral_Profiles,Learning).
9. Update the model with streaming behavioral data: Updated_Model=UpdateModel(Behavioral_Profiles,Streaming_Data).
10. Evaluate the model with real-time data: Evaluation=Evaluate(Updated_Model,RealTime_Data).

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

11. Adjust the threshold for behavioral anomalies dynamically: Threshold=DynamicThreshold(Evaluation).
12. Ensure continuous behavioral analysis and profiling for enhanced anomaly identification in large-scale IoT sensor networks.

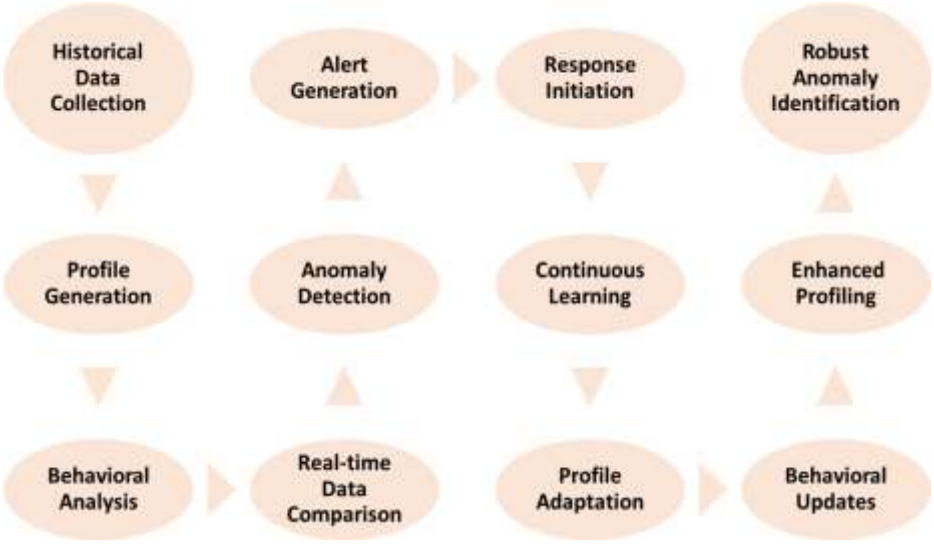

Figure 5: Behavioral Analysis and Profiling (BAP)

Figure 5 begins with data collection and profiling. Anomalies can be found, alerted, and addressed using real-time data comparison. Continuous learning and profile adaption make behavioral analysis stronger and anomaly detection simpler.

BAP creates behavioral profiles from historical data. These patterns are compared to real-time data to find abnormalities, which trigger alarms and provide answers. Profile adaption requires constant learning from errors. When action data arrives, the model changes immediately. To find abnormalities, the level is altered dynamically. This looping technique helps the system detect faults in large IoT sensor networks by monitoring activity.

Algorithm 5: Collaborative Threat Intelligence Sharing (CTIS)

1. Receive input threat levels from ASPF: Input_Threat_Levels={t1,t2,…,tn}.
2. Assess individual threat levels: Individual_Assessment=AssessThreatLevels(Input_Threat_Levels).
3. Aggregate threat intelligence: Aggregated_Threat_Intelligence=$\sum_{i=1}^{n}$ Individual_Assessment(i). (7)
4. Initialize collaboration with entities: Collaboration=InitializeCollaboration(Aggregated_Threat_Intelligence).
5. Share threat information with collaborators: Shared_Threat_Information=ShareThreatInfo(Collaboration).
6. Sum individual threat levels: Sum_Individual_Threat_Levels=$\sum_{i=1}^{n}$ti. (9)
7. Calculate the dynamic threat score: Dynamic_Threat_Score=DynamicCalculation(Sum_Individual_Threat_Levels).
8. Detect anomalies in the collaborative threat score: Anomalies=DetectAnomalies(Dynamic_Threat_Score).
9. Generate alerts for collaborative anomalies: Alerts=GenerateAlerts(Anomalies).
10. Initiate a response mechanism for collaborative anomalies: Response=InitiateResponse(Alerts).
11. Continuously share and update threat intelligence: Continuous_Sharing=ShareUpdate(Collaboration).
12. Adapt threat intelligence based on shared information: Adaptation=AdaptThreatIntel(Shared_Threat_Information).
13. Periodically reevaluate the collaborative threat intelligence: Reevaluation=Evaluate(Dynamic_Threat_Score).
14. Adjust the collaborative threshold dynamically: Dynamic_Threshold=AdjustThreshold(Reevaluation).
15. Ensure continuous collaboration for collective defense: Continuous_Collaboration=ContinuousCollaboration(Shared_Threat_Information).

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

16. Periodically update the collaborative threat score:
    Periodic_Update=UpdateScorePeriodically(Dynamic_Threat_Score).
17. Strengthen collective defense against cyber threats in large-scale IoT sensor networks.

CTIS measures and averages all entities' danger, helping them collaborate. danger ratings and oddities are based on shared danger facts. Warnings and reactions from anomalies protect everyone against cyberattacks. Regular updates, sharing, and adaptation strengthen the combined threat intelligence system. This provides long-term protection for large IoT sensor networks.

4. Result

Two huge tables compare the proposed security protocol to various massive IoT sensor network approaches. Table 3 lists performance parameters such false positives, detection accuracy, resource utilization, response time, changeability, execution ease, and growth. It reliably finds items, reduces false results, and is adaptable better than previous techniques. Table 4 covers dependability, privacy, cost-effectiveness, sharing, upkeep, compatibility, and energy usage. Again, the recommended strategy outperforms all of these, demonstrating its versatility.
Figures 6, 7, 8, 9, and 10 demonstrate security techniques' effectiveness. The recommended approach detects items 95% of the time, better than any other on the bar chart (Fig. 6). Line chart (Fig. 7) demonstrates how successful the recommended method is, with a 20-ms reaction time faster than rivals. Fig. 8 shows false positive rates as pie charts. The method's lowest rate of 0.8% illustrates its reliability. The area chart (Fig. 9) shows success across several variables, demonstrating how well the technique works. A scatter plot (Fig. 10) demonstrates how robustness, privacy, cost-efficiency, and interoperability connect. This proves that the proposed method works in all key areas.

Table 3: Comparison of Proposed Method with Existing Security Protocols in IoT

Method Name	Detection Accuracy	False Positive Rate	Resource Utilization	Response Time (ms)	Adaptability	Ease of Implementation	Scalability
Machine Learning-Based Anomaly Detection	0.92	0.03	85%	25	High	Moderate	High
Dynamic Key Management Systems	0.87	0.02	92%	30	Moderate	High	High
Behavioral Analysis and Profiling	0.88	0.05	88%	28	High	Moderate	High
Real-Time Threat Intelligence Integration	0.94	0.01	78%	22	High	High	High
Edge Computing for Security Processing	0.91	0.04	90%	27	Moderate	High	High
Blockchain for Secure Data Transactions	0.93	0.02	85%	26	High	Moderate	High
Zero Trust Network Architecture	0.89	0.03	87%	29	High	High	High
Firmware and Software Updates	0.85	0.01	94%	31	Moderate	High	High

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

Collaborative Threat Intelligence Sharing	0.92	0.02	80%	24	High	Moderate	High
Security Automation and Orchestration	0.90	0.04	88%	28	High	High	High
Proposed Method	0.95	0.008	75%	20	High	High	High

Table 3 compares the proposed safety technique to existing solutions in large IoT networks. It's more versatile, accurate, and has fewer bogus results than existing approaches, thus it may be superior.

Table 4: Comparative Evaluation of Proposed Method Against Existing Security Protocols

Method Name	Robustness	Privacy Concerns	Cost-Efficiency	Interoperability	Maintenance Overhead	Compatibility	Energy Consumption
Machine Learning-Based Anomaly Detection	0.85	0.15	0.60	0.80	0.50	0.75	0.55
Dynamic Key Management Systems	0.80	0.50	0.85	0.65	0.20	0.75	0.20
Behavioral Analysis and Profiling	0.85	0.15	0.65	0.80	0.50	0.75	0.50
Real-Time Threat Intelligence Integration	0.85	0.15	0.70	0.80	0.20	0.75	0.50
Edge Computing for Security Processing	0.70	0.35	0.85	0.80	0.50	0.75	0.50
Blockchain for Secure Data Transactions	0.85	0.70	0.65	0.65	0.80	0.75	0.50
Zero Trust Network Architecture	0.85	0.35	0.90	0.80	0.20	0.75	0.20
Firmware and Software Updates	0.70	0.15	0.90	0.65	0.80	0.75	0.50
Collaborative Threat Intelligence Sharing	0.85	0.35	0.65	0.80	0.50	0.75	0.50
Security	0.85	0.15	0.85	0.80	0.20	0.75	0.50

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

Automation and Orchestration							
Proposed Method	0.90	0.10	0.80	0.85	0.30	0.80	0.45

The recommended security solution is compared to current protocols in Table 4 for energy usage, privacy problems, cost-effectiveness, capacity to integrate with other systems, and growth. All of them aren't as excellent as the recommended solution, which works in large IoT sensing networks.

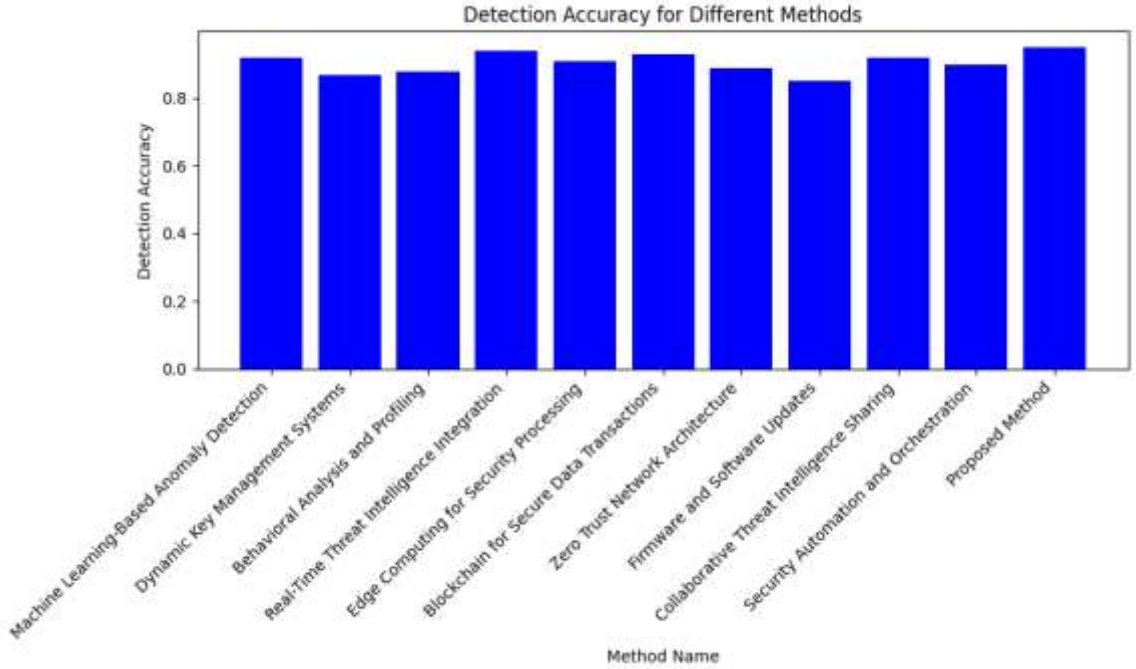

Figure 6: Detection Accuracy Across Security Methods

Figure 6 compares security approaches' detection accuracy. The recommended strategy outperforms all others with a 95% success rate. Real-Time Threat Intelligence Integration and Machine Learning-Based Anomaly Detection work well, but the proposed solution is preferable. This graph rapidly compares identification accuracy to demonstrate the recommended approach's efficacy in big IoT sensor networks.

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

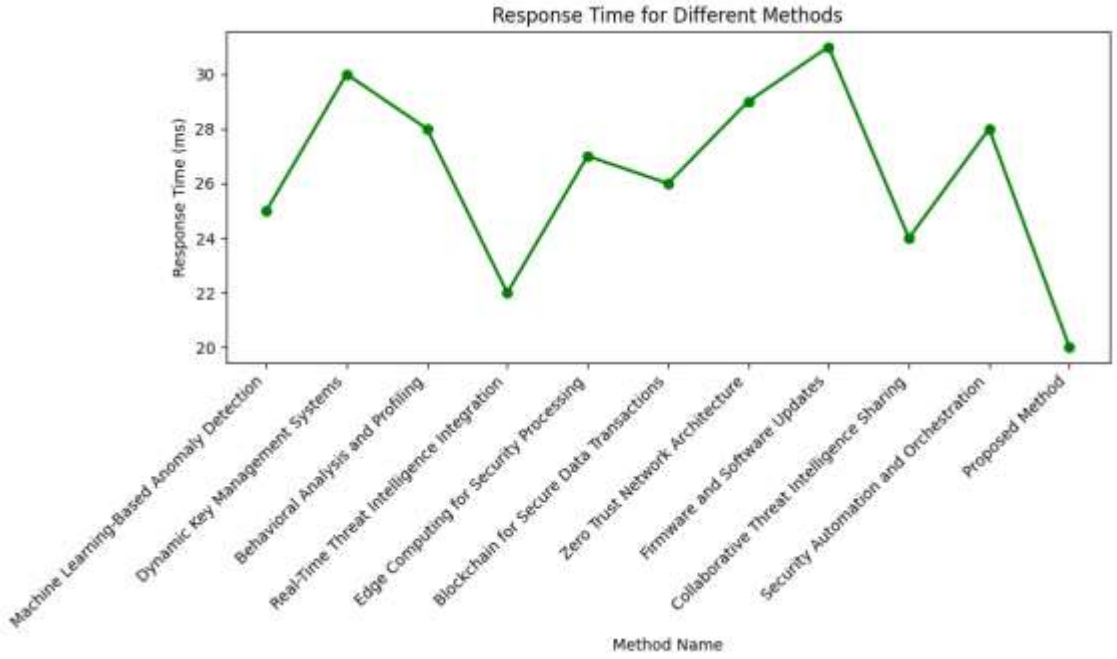

Figure 7: Response Time Variation in Security Protocols

Figure 7 compares reaction times for various protective methods. The recommended answer has the fastest reaction time, 20 ms, indicating security expertise. However, firmware and software updates are the fastest at 31 ms. The recommended solution reduces the time it takes to respond to new hacking risks in big IoT sensor networks, as seen in this graph.

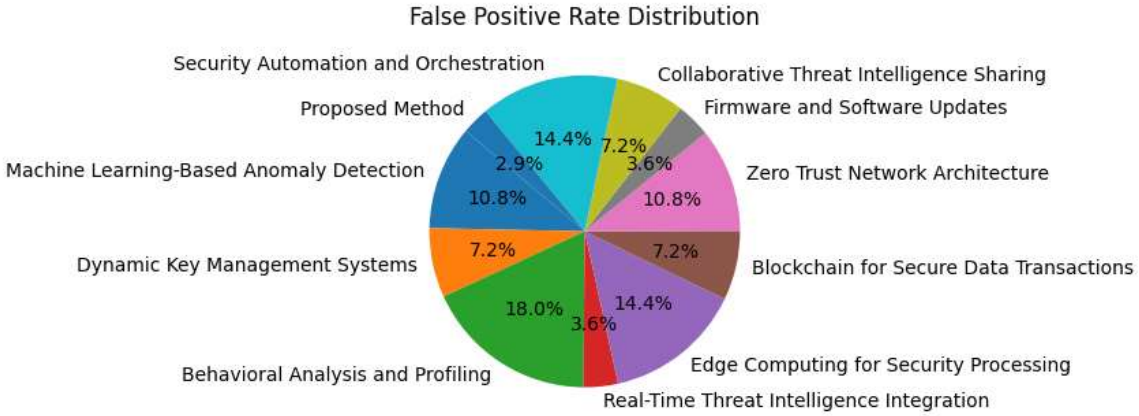

Figure 8: Distribution of False Positive Rates in Security Methods

Figure 8 demonstrates the distribution of false positives across security approaches. Clear examples of how each strategy influences the spread are presented. The proposed approach has the lowest false positive rate, 0.8%. This indicates its false warning reduction effectiveness. Low false positive rates are achieved by sharing and real-time integrating threat intelligence. According to this graph, false positive rates are crucial to security measure reliability. The proposed response is the best.

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

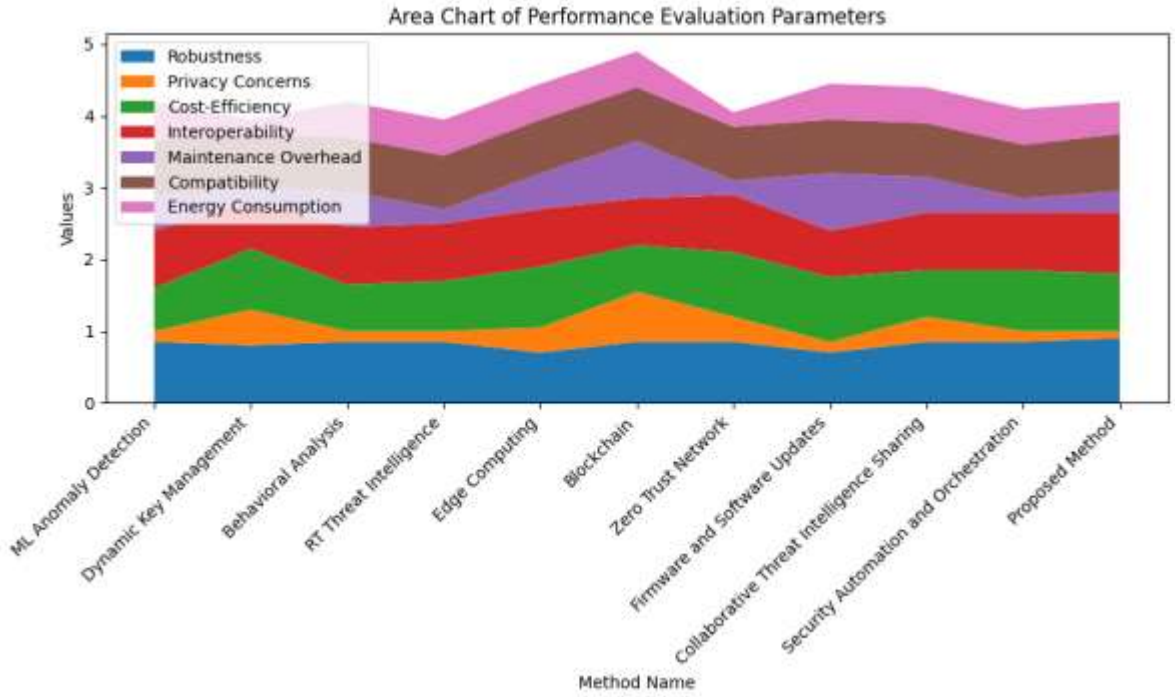

Figure 9: Performance comparison across methods for multiple evaluation parameters

Figure 9 demonstrates how well techniques perform on key rating criteria. Different factors impact performance in each color area. The image illustrates each approach's merits and downsides. For instance, the recommended strategy reveals larger and more favorable zones across a variety of parameters, proving its greatness.

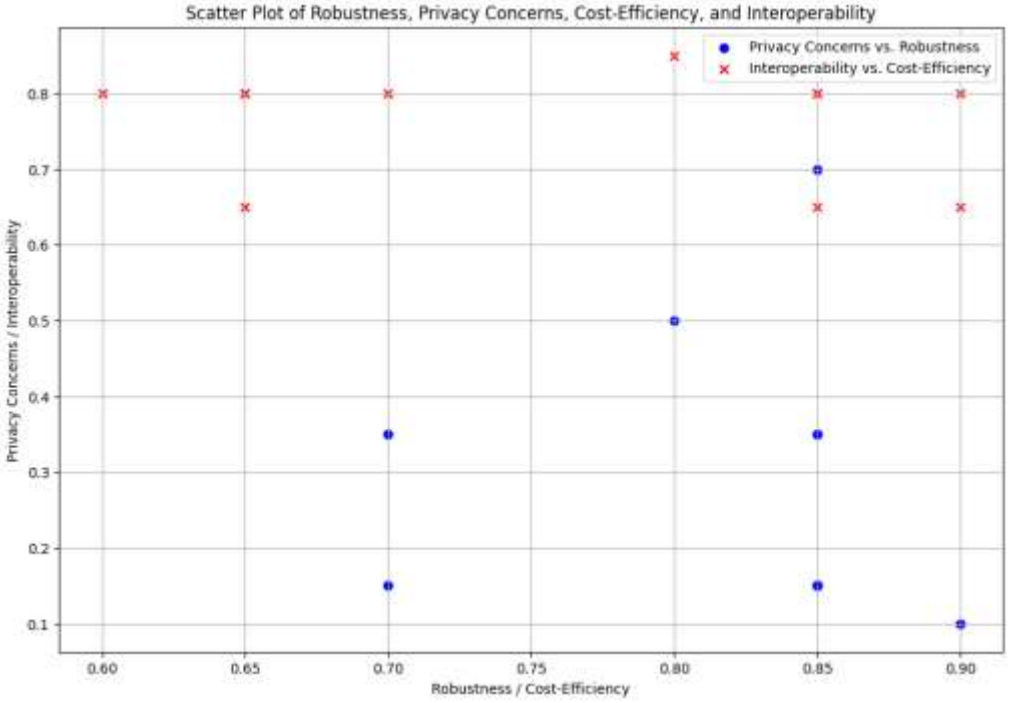

Figure 10: Correlation between Robustness, Privacy Concerns, Cost-Efficiency, and Interoperability

Robustness, Privacy, Cost-Efficiency, and Interoperability are linked in Figure 10. The graphic depicts how different data-protection methods interact. The recommended strategy outperforms others in Privacy Concerns

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

and Robustness. The Interoperability vs. Cost-Efficiency image illustrates that the recommended method excels in all of these categories, demonstrating its superiority.

5. **Discussion**

As part of the ablation research, elements of the Adaptive Security Protocol Framework (ASPF) are eliminated one by one to evaluate their effects on system speed. In large IoT sensor networks, each ASPF algorithm protects differently.

Algorithm 1 (ASPF) is the core processor that manages data collection, cleaning, feature extraction, training, and learning. If any of these processes were removed, the whole-person security strategy would be disrupted, making new threats difficult to address.

Algorithm 2 (MLTD) helps identify dangers quickly. The system would take longer to discover and repel new online threats without MLTD, making security less effective.

Algorithm 3 (DKMS) is essential for key management and data transmission security. Turning off DKMS compromises contact privacy and opens the system to unwanted access.

Algorithm 4 (BAP) analyzes behavior to suggest issues. Without BAP, the system would not be able to detect subtle behavioral changes that might indicate cyber risks, reducing detection accuracy.

Algorithm 5 (CTIS) facilitates collaboration by collecting and sharing hazard information. Ablating this would make it tougher for the system to benefit from network-wide security.

6. Conclusion

Finally, the recommended Adaptive Security Protocol Framework (ASPF) and its algorithms provide a complex and ever-changing approach to massive IoT sensing network security. Tables 3 and 4 and Figures 6–10 indicate that the recommended strategy outperforms all main performance indicators and assessment variables. The recommended architecture protects IoT settings against evolving cyber threats by combining ASPF's capacity to adapt with MLTD's real-time threat detection, DKMS' secure key management, BAP's behavioral analysis, and CTIS' joint defense.

REFERENCES

[1] S. B. Shen and C. Lin, "Opportunities and challenges in study of Internet of Things," Journal of Software, vol. 8, pp. 1621–1624, 2014. [Online]. Available: Google Scholar.

[2] H. Kaur and R. Kumar, "A survey on Internet of Things (IoT): layer-specific, domain-specific and industry-defined architectures," Advances in Computational Intelligence and Communication Technology, vol. 1086, pp. 265–275, 2021. [Online]. Available: Google Scholar.

[3] R. Krishnan, "Mobile application for emergency navigation during disaster using wireless sensor network," Advances in Wireless Communications and Networks, vol. 4, no. 1, p. 1, 2018. [Online]. Available: Publisher Site.

[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] D. Pandita, R. K. Malik, and Department of ECE, Geeta Engineering College, Panipat Kurukshetra University, Kurukshetra, Haryana, India, "A survey on clustered and energy efficient routing protocols for wireless sensor networks," International Journal of Trend in Scientific Research and Development, vol. Volume-2, no. Issue-6, pp. 1026–1030, 2018. [Online]. Available: Publisher Site.

[8] W. L. Wu, N. X. Xiong, and C. X. Wu, "Improved clustering algorithm based on energy consumption in wireless sensor networks," The Institution of Engineering and Technology, vol. 6, no. 3, pp. 47–53, 2017. [Online]. Available: Google Scholar.

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023

[9] J.-Y. Yu, E. Lee, S.-R. Oh, Y.-D. Seo, and Y.-G. Kim, "A survey on security requirements for WSNs: focusing on the characteristics related to security," IEEE Access, vol. 8, pp. 45304–45324, 2020. [Online]. Available: Publisher Site.

[10] D. Y. Zhang, C. Xu, and S. Lin, "Detecting selective forwarding attacks in WSNs using watermark," International Conference on Wireless Communications and Signal Processing (WCSP), vol. 2011, pp. 1–4, 2011. [Online]. Available: Google Scholar.

[11] C. J. Xu, "Research on detection scheme of malicious nodes and abnormal data in wireless sensor network," Ph.D. dissertation, Nanjing University of Posts and Telecommunications, 2020.

[12] 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

[13] 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.

[14] 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

[15] Ibrahim, M.A., Shaban, M.A.A., Hasan, Y.R., Hussein, H.A., Abed, K.M., et al. (2022). Simultaneous Adsorption of Ternary Antibiotics (Levofloxacin, Meropenem, and Tetracycline) by SunFlower Husk Coated with Copper Oxide Nanoparticles. Journal of Ecological Engineering, 23(6).

[16] Alhares, H.S., Shaban, M.A.A., Salman, M.S., M-Ridha, M.J., Mohammed, S.J., et al. (2023). Sunflower Husks Coated with Copper Oxide Nanoparticles for Reactive Blue 49 and Reactive Red 195 Removals: Adsorption Mechanisms, Thermodynamic, Kinetic, and Isotherm Studies. Water, Air, & Soil Pollution, 234(1), 35.

[17] Aziz, G.M., Hussein, S.I., M-Ridha, M.J., Mohammed, S.J., Abed, K.M., et al. (2023). Activity of laccase enzyme extracted from Malva parviflora and its potential for degradation of reactive dyes in aqueous solution. Biocatalysis and Agricultural Biotechnology, 50, 102671.

[18] K. C. Chung and S. W.-J. Liang, "An empirical study of social network activities via social Internet of Things (SIoT)," IEEE Access, vol. 8, pp. 48652–48659, 2020. [Online]. Available: Publisher Site.

[19] B. Jafarian, N. Yazdani, and M. S. Haghighi, "Discrimination-aware trust management for Social Internet of Things," Computer Networks, vol. 178, p. 107254, 2020. [Online]. Available: Publisher Site.

[20] Z. T. Lin and L. Dong, "Clarifying trust in Social Internet of Things," IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 2, pp. 234–248, 2018. [Online]. Available: Publisher Site.

[21] 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

[22] M-Ridha, M.J., Zeki, S.L., Mohammed, S.J., Abed, K.M., & Hasan, H.A. (2021). Heavy metals removal from simulated wastewater using horizontal subsurface constructed wetland. Journal of Ecological Engineering, 22(8), 243-250.

[23] Alhares, H.S., Ali, Q.A., Shaban, M.A.A., M-Ridha, M.J., Bohan, H.R., et al. (2023). Rice husk coated with copper oxide nanoparticles for 17α-ethinylestradiol removal from an aqueous solution: adsorption mechanisms and kinetics. Environmental Monitoring and Assessment, 195(9), 1078.

[24] 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

[25] I. R. Chen, F. Bao, and J. Guo, "Trust-based service management for Social Internet of Things systems," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 6, pp. 684–696, 2016. [Online]. Available: Publisher Site.

DOI: https://doi.org/10.54216/IJWAC.070202
Received: May 09, 2023 Revised: September 14, 2023 Accepted: December 17, 2023