International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Securing Industrial IoT Environments through Machine Learning-Based Anomaly Detection in the Age of Pervasive Connectivity

[1]Bassam Mohammad Elzaghmouri*,[2]Ahmad Khader Habboush, [3]Marwan Abu-Zanona , [4]Suprava Ranjan Laha, [5]Binod Kumar Pattanayak, [6]Saumendra Pattnaik, [7]Bibhuprasad Mohanty

Submitted: 19/09/2023 **Revised**: 16/11/2023 **Accepted**: 30/11/2023

Abstract: In an era characterized by the relentless evolution of Internet of Things (IoT) technologies, marked by the pervasive adoption of smart devices and the ever-expanding realm of Internet connectivity, the IoT has seamlessly integrated itself into our daily lives. This integration has ushered in a new era for manufacturing companies, enabling them to conduct real-time monitoring of their machinery, supervise product quality, and closely monitor environmental variables within their facilities. In addition to the immediate benefits of risk mitigation and loss prevention, this multifaceted approach has provided decision-makers with a comprehensive perspective for making informed decisions. People are now more dependent than ever in IoT devices and services. However, anomalies within IoT networks pose a critical concern despite the IoT's immense potential. These anomalies can pose significant security and safety risks if they go undetected. Identifying and alerting users of these anomalies on time has become crucial for preventing potential damages and losses. In response to this imperative, our research endeavors to utilize the power of Machine Learning and Deep Learning techniques to detect anomalies in IoT networks. We undertake exhaustive experiments with the IoT-23 dataset to validate our methodology empirically. Our research examines an exhaustive comparison of numerous models, assessing their performance and time efficiency to determine the optimal algorithm for achieving high detection accuracy under strict time constraints. This research represents an important step towards enhancing the security of Industrial IoT environments, thereby protecting vital infrastructure and ensuring the integrity of industrial operations in our increasingly interconnected world.

Keywords: *Industrial IoT, Anomaly detection, Security, Deep Learning, Risk mitigation .*

1. Introduction

The Internet of Things (IoT) represents a paradigm shift comparable to the advent of the Internet in the global information industry. It is an intelligent network facilitating communication and data transmission between internet-connected devices [3]. The Internet of Things has allowed humanity to trace, monitor, locate, identify, and manage a vast array of objects [4]. In the wake of the Internet and mobile device revolutions, IoT has arisen as a dynamic and intensively researched area of computer science. The number of IoT devices connected to the Internet increases annually in numerous industries, including Smart Healthcare, Smart Transportation, Smart Governance, Smart Agriculture, Smart Grid, Smart Home, and Smart Supply Chain [5, 6]. IoT's convenience has altered human behavior, especially among younger generations who have enthusiastically adopted IoT services, from smart bulbs and furnaces to refrigerators, air conditioners, temperature sensors, and smoke detectors [7]. Nonetheless, expanding privacy and security concerns have accompanied this remarkable development. As more and more devices connect to the Internet, the attack surface expands, giving malicious actors more opportunities to access sensitive data. These interconnected devices frequently collect and store personal information without the user's understanding of IoT technology, making them susceptible to data theft and even remote control by hackers [8]. This impedes the development of IoT technology and the growth of its infrastructure.

Consequently, ensuring the security & privacy of these pervasively & extensively interconnected devices has become a formidable obstacle. Moreover, given the limitations of conventional data capture, storage, and

[1]*Department of Computer Science .Faculty of Information technology, Jerash University, Jerash, Jordan, Email:b.el-zaghmouri@jpu.edu.jo*
[2]*Faculty of Information Technology, Jerash University, Jerash, Jordan, Email: ahmad_ram2001@jpu.edu.jo*
[3]*Department of Management Information Systems, College of Business Administration, King Faisal University, Saudi Arabia, Email: mabozanoneh@kfu.edu.sa.*
[4]*Department of Computer Science and Engineering, Institute of technical Education and Research, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India, Email: supravalaha@gmail.com*
[5]*Department of Computer Science and Engineering, Institute of technical Education and Research, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India, Email: binodpattanayak@soa.ac.in*
[6]*Department of Computer Science and Engineering, Institute of technical Education and Research, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India, Email: saumendrapattnaik@soa.ac.in*
[7]*Department of Electronics and Communication Engineering, Institute of technical Education and Research, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India, Email: bibhumohanty@soa.ac.in*

processing techniques [9], managing the colossal volume of data generated by these devices represents a formidable challenge.

Due to advancements in ML & DL learning algorithms can now adapt and enhance their performance by acquiring knowledge from trained data. A trained learning algorithm can distinguish between normal, benign network traffic and malicious activity, enabling the detection of aberrant network behavior and preventing unauthorized access. These learning algorithms are classified broadly as supervised-learning & unsupervised-learning. To improve security in IoT environments, we employ lightweight machine learning and neural network models to improve the accuracy of identifying malicious nodes. The core of our model collects IoT traffic data and transmits it to trained ML and DL models for evaluation. This strategy enables us to select the model that best meets users' diverse demands and requirements. The enormous volume of data circulating in IoT networks, coupled with the inherent heterogeneity of this data, presents challenges for enhancing security while meeting diverse requirements, such as cost-effectiveness, dependability, and performance [10]. It is essential to balance security measures and their prospective impact on factors such as cost and latency. In addition, the proliferation of connected devices increases the attack surface, making it essential for devices, even those with limited resources such as smart bulbs, to detect intrusions with minimal complexity and time. Machine Learning (ML) and Deep Learning (DL) techniques provide an efficient method for reducing complexity by learning from trained data, thereby facilitating intrusion detection. The growing privacy and security concerns inherent to the Internet of Things have prompted researchers to develop frameworks for automatic IoT sensor attacks & anomaly detection [11, 12].

This paper proposes using ML and DL algorithms for anomaly detection, including Support Vector Machines, Decision Trees, Naive Bayes, and Convolutional Neural Networks. Our evaluation, based on precision and time requirements, enables us to determine which algorithm is most appropriate for deployment. We use the IoT-23 dataset as the basis for our ML and DL implementations, thereby contributing to ongoing efforts to strengthen the security of Industrial IoT environments.

2. Proposed Approach

Our proposed method introduces a model for an anomaly detection system meant to improve IoT security. As shown in Figure 1, our model consists of several essential components.

Traffic Capture Unit: Initially, a dedicated traffic capture unit is deployed to intercept and collect traffic flow emanating from IoT sensors and direct it to the central unit for analysis [13].

Central-Unit: The captured traffic flow is then sent to a central computing device, which may be located in a local computing environment or a cloud-based infrastructure.

ML & DL Models: Various Machine Learning (ML) and Deep Learning (DL) models are executed within the central unit to evaluate their performance and computational costs. These models are indispensable for anomaly detection [14].

Database Integration: The central unit simultaneously archives the captured traffic flow in its database, facilitating future studies and model recalibrations. This database is a valuable resource for continuous analysis and enhancement.

Model Selection: After evaluating the performance and cost of ML/DL models, users or system administrators are empowered to select the most appropriate model for anomaly detection. The choice may be guided by factors such as accuracy and processing time, which vary according to the specific user's requirements and circumstances.

Anomaly Detection: In the active anomaly detection phase, the central unit analyses incoming traffic utilizing the selected ML/DL model. Upon detecting anomalies, the central unit initiates a series of actions, including packet dropping, IP address blacklisting, user alerts, physical inspections, and other pertinent responses.

Customization and Adaptation: Given the diversity of IoT security scenarios, our model emphasizes allowing users to select the ML/DL model that best meets their specific requirements. In addition, the continuous accumulation of traffic flow data within the database enables the generation of new datasets for future recalibration of existing ML/DL models, further improving their performance.

We investigate numerous Ml algorithms, including Support Vector Machine (SVM), Naive Bayes, & Decision Tree. Simultaneously, we investigate Deep Learning techniques, specifically Convolutional Neural Networks (CNN). After training these algorithms with the collected data, exhaustive computations are performed to detect anomalies in the IoT system. This operation can be carried out locally on dedicated hardware or in the cloud.

To demonstrate the effectiveness of our method, we divide the dataset into training and testing data subsets. We draw conclusions based on the outcomes of the analysis using the trained algorithms. If an anomaly is detected, various response actions can be triggered, such as packet dropping, IP address blacklisting, user alerts, physical inspections, malware scanning, and other appropriate actions. This multifaceted strategy ensures that IoT security is maintained and anomalies are promptly and thoroughly addressed.

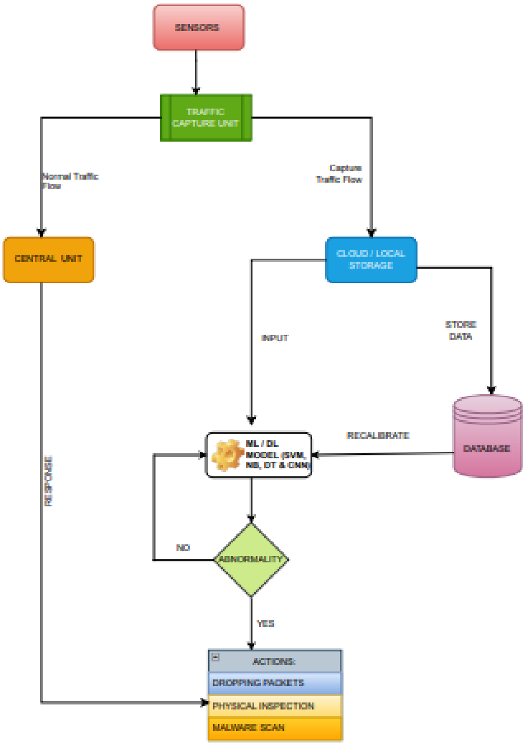

Fig. 1. IoT Security based Anomaly Detection System

We thoroughly compare the results obtained from the numerous methods under consideration. Our primary concentration is on two crucial factors: the "accuracy" and "time cost" of each algorithm. Precision evaluation is crucial because it directly influences the dependability of anomaly detection. However, time is a valuable resource in the context of IoT networks. Consequently, the time cost incurred by each algorithm is of critical importance. For instance, a model with 100 percent accuracy that requires extensive processing time may not be suitable for IoT networks. This consideration is especially pertinent due to the limited resources of IoT devices.

Therefore, our proposed model is flexible enough to accommodate various user scenarios. It seeks to provide an optimal solution corresponding to various users' unique requirements and resources. Whether it is a large corporation with abundant resources seeking the highest levels of precision or a smaller business focusing on cost-effectiveness, our approach offers a customized solution to meet their specific needs. Our model ensures that IoT security remains effective and efficient across a broad spectrum of applications and organizations by allowing users to make informed decisions based on the trade-off between accuracy and time cost.

2.1. Dataset

In this study, we obtained our data from the IoT-23 dataset, a relatively recent dataset that was released in January 2020. This contemporary dataset consists of network traffic information collected from three smart home IoT devices: Amazon Echo, Philips HUE, and Somfy Door Lock. IoT-23 is a scrupulously curated repository of real-world IoT malware infections and benign traffic for developing Machine Learning algorithms. The IoT-23 data set contains 23 recordings, also known as scenarios. Twenty of these recordings pertain to malicious network activity, while the remaining three depict benign network behavior. Each scenario involving infected devices is labeled with the potential name of the malware sample executed during that particular instance. The dataset includes various malware labels, categorizing the IoT-23 captures into different classes such as Attack, C&C (Command and Control), C&C-File Download, C&C-Heartbeat, C&C-Heart Beat-Attack, C&C-Heart Beat-File Download, C&C-Mirai, C&C-Torii, DDoS (Distributed Denial of Service), File Download, Okiru, & Okiru-Attack. In addition, the IoT-23 dataset was subjected to network analysis using the Zeek network analysis software. We used the conn.log.labeled format for the IoT-23 dataset, which represents the Zeek conn.log file generated from the original pcap file by the Zeek network analyzer. We adopted the strategy of extracting a subset of records from each dataset and then merging them into a new dataset due to the large dataset. This approach was adopted to ensure that our computing resources could effectively handle the workload of the new dataset. Importantly, this new dataset retains the majority of attack types from the original IoT-23 dataset, allowing us to effectively conduct our research while concentrating on the pertinent IoT security aspects.

2.2 Dataset Preprocessing

The IoT-23 dataset was preprocessed methodically for analysis. Here are our data preprocessing steps:

I. Data Loading: Initially, we loaded all 23 individual datasets from the IoT-23 Dataset using the Python library Pandas. Each dataset was imported into its data frame. Notably, we employed a condition to skip the initial 10 rows and instead read the following 100,000 rows.

II. Combination of Dataset: Subsequently, we merged all 23 data frames into a single exhaustive data frame, facilitating a unified analysis.

III. Feature Selection: To improve the efficiency and relevance of our analysis, we eliminated variables that did not substantially influence the results from the dataset. ts (timestamp), uid (user ID), id. orig h (original host ID), id.orig p (original port), id. resp h (response host ID), id. resp p (response port), service, local orig, local resp, and history were non-impactful variables.

IV. Handling Categorical Variables: To facilitate their incorporation in our analysis, we transformed categorical variables such as proto (protocol) and conn state (connection state) into dummy variables.
V. Handling missing values: We dealt with absent values by replacing them with zeros, thereby ensuring a complete and consistent dataset.
VI. Generated Dataset: The processed and combined dataset was saved as the "iot23_combined.csv" file, which will be the foundation for our subsequent analysis.

The output file "iot23_combined.csv" comprises **11, 55, 873** records in total. As detailed in Table 1, this combined data set contains 09 distinct varieties of attacks, including Okiru (O), DDoS, Attack (A), D&D-HeartBeat (HB), D&D-FileDownload (FD), D&D-Torii (T), FileDownload (FD), D&D-HeartBeat-FileDownload (HBFD), and D&D-Mirai (M).

We divided the combined dataset into two subsets for validation: a training dataset containing 80% of the data and a testing dataset containing the remaining 20%. This division allows us to evaluate the effectiveness and precision of our anomaly detection models.

Table 1 Total IOT23 attack types

Varieties of labels	Count
D&D-M	1
D&D-HBFD	8
FD	15
D&D-T	38
D&D-FD	68
D&D-HB	542
A	4820
DDoS	159886
Benign	263650
O	726845

3. Experimentation and Result Analysis

This section provides an in-depth analysis of the outcomes produced by the employed algorithms. Our analysis includes the presentation of each algorithm's confusion matrix, which provides insight into its efficacy and the computation time necessary for anomaly detection..

3.1. Hardware setup

Experiments were conducted on a computer with the following specifications: a 3.30 GHz Intel Core 7700k processor with 4 GB of RAM operating at 3220 MHz. In addition, the analysis was conducted on Windows 7 using the Anaconda Jupyter Notebook platform, Python version 3.8, and the Tensor flow 2.4 framework..

3.2. Experimentation

To measure how well the model works, we use several key metrics, each of which has a different purpose:

I. Time: This indicator quantifies an algorithm's time to execute a certain ML/DL model. Due to resource constraints, methods with high computing demands may not work in IoT contexts.

II. True positives: The model adequately predicts the positive class, demonstrating its anomaly detection skill.

III. False Positives: In contrast, this metric represents instances in which the model inaccurately predicts the positive class, signifying instances in which it incorrectly identifies anomalies.

IV. Precision: Precision is a crucial metric that quantifies the correct identification of positive instances within the model. The following Equation 1 determines it:

$$Precision = \frac{True\_Positives}{True\_Positives + False\_Positives} \quad (1)$$

Precision provides insight into the model's ability to avoid false alarms and precisely identify natural anomalies, making it a valuable metric for evaluating anomaly detection.

V. Recall: It is also called sensitivity and quantifies the model's capacity to correctly identify the number of positive instances shown in Equation 2. Recall is a crucial metric that measures the model's ability to identify and correctly classify all positive instances, thereby minimizing the risk of missing potential anomalies.

$$Recall = \frac{True\_Poditives}{True\_Positives + False\_Negatives} \quad (2)$$

VI. F1 Score: The harmonic mean of precision and recall is calculated using the F1 score, a comprehensive statistic considering false positives and negatives. Due to its fair evaluation, this metric is frequently preferred. The F1 score establishes a balance between precision and recall, comprehensively evaluating a model's anomaly detection performance, as shown in Equation 3.

$$F1 = 2 * \frac{Precision * recall}{Precision + recall} \quad (3)$$

VII Support Score: The Python library Sci-kit-learn measures support score. It shows how many times each label is true. This statistic quantifies the frequency or prevalence of each label in the dataset, aiding classification and analysis.

3.3. Result Analysis

Several noteworthy findings emerged from analyzing our test results for Machine Learning (ML) and Deep Learning (DL) techniques. Different ML/DL models exhibited varying degrees of accuracy, precision, recall, and F1 score, casting light on their respective strengths and weaknesses. These differences have significant implications for the applicability of each technique in the context of IoT anomaly detection. In addition, the computational time required by each algorithm was evaluated to determine their applicability in real-time IoT environments. These test results allow us to make informed decisions regarding selecting ML/DL methods, balancing accuracy, computational efficiency, and the requirements of various IoT security scenarios. The results provide vital insights for enhancing IoT environment security.

The supervised learning technique used in our work is based on Bayes' theorem and is primarily used for classification tasks in which predictions are based on probabilities. It is well-known for its simplicity and efficacy in machine-learning model construction. However, as presented in Table 2, our findings indicate that the Naive Bayes algorithm attained an overall accuracy of only 51% with a relatively quick execution time of 7 seconds. Sadly, it had the lowest accuracy of all the techniques evaluated in our study. In the context of our IoT anomaly detection task, these results highlight the limitations of the Naive Bayes algorithm. While it is well-known for its simplicity, its performance in this scenario must improve, compelling us to investigate alternative techniques with greater precision for enhancing security in IoT environments.

Table 2. Metrics for NBs Algorithm

Metrics	Precision	Recall	F1 Score	Support
Accuracy			0.51	377824
Macro Average	0.65	0.71	0.48	377824
Weighted Average	0.8	0.51	0.41	377824

The SVM algorithm effectively classifies data points by pursuing an optimal hyperplane dependent on the number of features. This hyperplane categorizes data points based on their position relative to it. SVM leverages extreme data points, known as support vectors, to maximize the margin of the classifier, thereby enhancing classification performance. Nevertheless, as shown in Table 3, the overall accuracy obtained by SVM is only 85%. Although we can delve into the precision of each attack, it is notable that the execution time for SVM is approximately two hours. Although SVM achieves an accuracy level comparable to Decision Trees and Convolutional Neural Networks (CNN), it requires the longest computational time of all the methods analyzed in this study. This demonstrates the relationship between SVM's classification performance and computational requirements, prompting us to consider time efficiency when selecting an algorithm for IoT anomaly detection in resource-constrained environments.

Table 3 Metrics for SVM Algorithm

Metrics	Precision	Recall	F1 Score	Support
Accuracy			0.85	94325
Macro Average	0.49	0.42	0.41	94325
Weighted Average	0.8	0.85	0.65	94325

Supervised Machine Learning classifiers like Decision Trees are used for classification. It views the dataset as nodes, leaves, and branches. Nodes represent dataset attributes, leaf nodes indicate outcomes, and branches represent classification decision rules. Our results in Table 4 reveal that the Decision Trees method has 93% accuracy. Interestingly, its execution takes only 3 seconds. Decision Trees outperform all other methods in our analysis with the highest accuracy and computing efficiency. Due to its accuracy and computational efficiency, Decision Trees is our chosen solution for IoT anomaly detection due to its strong performance.

Table 4. Metrics for Decision Tree

Metrics	Precision	Recall	F1 Score	Support
Accuracy			0.93	933227
Macro Average	0.83	0.71	0.71	933227
Weighted Average	0.97	0.93	0.85	933227

CNNs are deep learning models with minimum preprocessing and architecture inspired by human brain neural patterns. CNNs have convolutional, pooling, fully connected, and normalization layers. These layers collaborate to extract data features. CNNs' core convolutional layer has multiple hyper parameters, including input and output channels, padding size, and kernel width and height. The network's capacity to detect complex data patterns depends on these hyper parameters. Pooling layers reduce data dimensionality by combining neuron cluster outputs into a single neuron in the next layer. This dimensionality reduction reduces network computing complexity while preserving vital information.

In contrast, fully connected layers allow the network to share information by connecting each neuron to the previous layer. CNNs are particularly effective in image and pattern recognition tasks because they can recognize complex patterns and features from unprocessed data. Their structural closeness to brain neural patterns makes them useful deep learning and AI tools. We used the Rectified Linear Unit (ReLu) activation function for the dense layers, a linear function that outputs the input directly if the result is positive and 0 otherwise. This activation function selection facilitates feature extraction and nonlinear changes within the network. In contrast, the output layer employs the Softmax activation function, which is a logistic function to normalize the output into a probability distribution, allowing for multi-class classification. We used the Adam optimizer for optimization, a gradient descent-based optimization technique noted for its efficacy in training deep neural networks. Our suggested CNN model consists of a total of 3,523,551 trainable parameters. This reflects the network's ability to adapt and learn from data. Figure 2 shows that our CNN model obtained 79 percent testing accuracy with an execution duration of around 4 minutes. While this accuracy is slightly lower than Decision Trees, it is worth noting that CNNs outperform Decision Trees when confronted with more complicated datasets and intricate patterns, demonstrating their applicability for applications requiring higher feature abstraction and recognition levels.

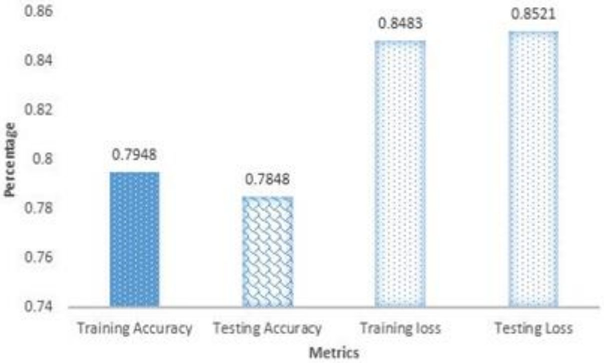

Fig. 2. Metrics for CNN Model

3.4. Comparison

Upon conducting a comparative analysis of our findings with previous research, it becomes apparent that our methodology has resulted in notable enhancements in the precision of specific algorithms Shown in figure 3. In the context of Naïve Bayes, our implementation demonstrated a notable accuracy of 0.51, surpassing the accuracy levels documented in two prior research [1] and [2], where accuracies of merely 0.23 were reported. This underscores the efficacy of our Naïve Bayes model in augmenting classification performance. Similarly, for the Support Vector Machine (SVM), our method achieved an accuracy of 0.85, outperforming the 0.67 and 0.69 accuracy values reported in studies [1] and [2], respectively. This observation showcases the exceptional efficacy of our Support Vector Machine (SVM) model in the context of classification tasks. Significantly, our Decision Trees (DT) model demonstrated a noteworthy accuracy of 0.93, highlighting its extraordinary categorization skills. The level of accuracy seen in this study dramatically exceeds the level recorded in the study [2], which documented an accuracy of 0.73. The Decision Trees approach we have developed is a reliable option for achieving precise classification outcomes.

However, our findings indicate significant enhancements in accuracy across many algorithms compared to previous research, confirming our methodology's efficacy in improving classification performance within the specific dataset. The presented picture provides a comprehensive representation of our research trajectory within the ever-evolving realm of Internet of Things (IoT) technologies, emphasizing its significance and influence on the security aspects of industrial IoT.

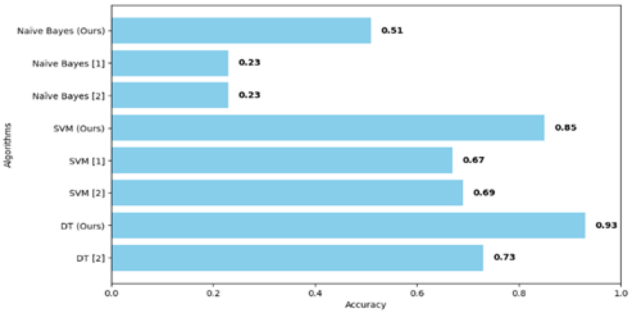

Fig 3: Comparison of our proposed Approach with the Existing Models

4. Conclusion and Future Scope

Our research has highlighted the crucial significance of machine learning-based anomaly detection in protecting Industrial IoT environments in an era characterized by the unrelenting expansion of Internet of Things (IoT) technology and the widespread integration of smart devices. As manufacturing companies embrace the potential of real-time monitoring, product quality supervision, and environmental variable control, they confront the inherent risks posed by anomalies within IoT networks. Anomalies that go undetected can pose significant security and safety risks, necessitating proactive measures.

Our research demonstrates the effectiveness of Machine Learning and Deep Learning techniques in detecting anomalies within IoT networks and alerting users. Through exhaustive experimentation with the IoT-23 dataset, we have validated our methodology's efficacy and compared various models in depth. This evaluation considered their performance and time efficiency, ultimately identifying the most effective algorithms capable of attaining high detection accuracy under strict time constraints.

In an era characterized by ubiquitous connectivity, our research represents a significant step toward enhancing the security of Industrial IoT environments. Future innovation and collaboration will protect crucial infrastructure and industrial activities in an increasingly interconnected world.

Acknowledgements

The researcher extends his sincere thanks and appreciation to the administration of King Faisal University in the Kingdom of Saudi Arabia for providing all forms of support to the university’s faculty members, especially in the field of scientific research. We extend our sincere thanks to the Deanship of Scientific Research at the university for agreeing to support this scientific research with GRANT ID 5311 .

Author contributions

Bassam Elzaghmouri : Conceptualization, Methodology, Software **Ahmad Habboush :** Field study, Data curation **Marwan Abu-Zanona :** Data curation, Writing-Original draft preparation, Software, Validation., Field study **Suprava Laha :** Visualization **Binod Pattanayak :** Writing-Reviewing and Editing **Saumendra Pattnaik :** Investigation **Bibhuprasad Mohanty:** Software, Validation

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Stoian, N. A. (2020). Machine learning for anomaly detection in IoT networks: Malware analysis on the iot-23 data set (Bachelor's thesis, University of Twente).

[2] Liang, Y., & Vankayalapati, N. Machine Learning and Deep Learning Methods for Better Anomaly Detection in IoT-23 Dataset Cybersecurity. Preprint.

[3] Laha, S. R., Pattanayak, B. K., & Pattnaik, S. (2022). Advancement of Environmental Monitoring System Using IoT and Sensor: A Comprehensive Analysis. AIMS Environmental Science, 9(6), 771-800.

[4] Pattnaik, S., Banerjee, S., Laha, S. R., Pattanayak, B. K., & Sahu, G. P. (2022). A Novel Intelligent Street Light Control System Using IoT. In Intelligent and Cloud Computing: Proceedings of ICICC 2021 (pp. 145-156). Singapore: Springer Nature Singapore.

[5] Mahapatra, S. K., Pattanayak, B. K., Pati, B., Laha, S. R., Pattnaik, S., & Mohanty, B. (2023). An IoT Based Novel Hybrid-Gamified Educational Approach to Enhance Student’s Learning Ability. International Journal of Intelligent Systems and Applications in Engineering, 11(3), 374-393.

[6] Laha, S. R., Mahapatra, S. K., Pattnaik, S., Pattanayak, B. K., & Pati, B. (2021). U-INS: an android-based navigation system. In Cognitive Informatics and Soft Computing: Proceeding of CISC 2020 (pp. 125-132). Springer Singapore.

[7] Biswal, A. K., Singh, D., Pattanayak, B. K., Samanta, D., Chaudhry, S. A., & Irshad, A. (2021). Adaptive fault-tolerant system and optimal power allocation for smart vehicles in smart cities using controller area network. Security and Communication Networks, 2021, 1-13.

[8] Hosenkhan, M. R., & Pattanayak, B. K. (2020). Security issues in internet of things (IoT): a comprehensive review. New Paradigm in Decision Science and Management: Proceedings of ICDSM 2018, 359-369.

[9] Mohi-Ud-Din, G., Zheng, J., Liu, Z., Asim, M., Chen, J., Liu, J., & Lin, Z. (2022, October). NIDS: Random Forest Based Novel Network Intrusion Detection System for Enhanced Cybersecurity in VANET's. In 2022 International Conference on Virtual Reality, Human-Computer Interaction and Artificial Intelligence (VRHCIAI) (pp. 255-260). IEEE.

[10] Sarker, I. H., Khan, A. I., Abushark, Y. B., & Alsolami, F. (2022). Internet of things (IoT) security intelligence: a comprehensive overview, machine learning solutions and research directions. Mobile Networks and Applications, 1-17.

[11] Sarker, I. H. (2022). Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects. Annals of Data Science, 1-26.

[12] Wu, B., Liu, S., Feng, R., Xie, X., Siow, J., & Lin, S. W. (2022). Enhancing security patch identification by capturing structures in commits. IEEE Transactions

on Dependable and Secure Computing.

[13] Thom, J., Thom, N., Sengupta, S., & Hand, E. (2022, January). Smart recon: Network traffic fingerprinting for IoT device identification. In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0072-0079). IEEE.

[14] Dushyant, K., Muskan, G., Annu, Gupta, A., & Pramanik, S. (2022). Utilizing Machine Learning and Deep Learning in Cybesecurity: An Innovative Approach. Cyber Security and Digital Forensics, 271-293.