Outlier Detection for IoT Frameworks Using Isolation Forest

V Lakshmi Chaitanya, M Sharmila Devi, Gaddam Anju Sree, Dudekula Aisha Thabasum, Uppu Sravani and Gandham Sneha
Department of Computer Science and Engineering, Santhiram Engineering College, Nandyal - 518501, Andhra Pradesh, India

Keywords: Outlier Detection, IoT Frameworks, Machine Learning, Wireless Networks, Anomaly Detection, Sensor Data, Network Security, Fault Detection, Data Management, Isolation Forest, One-Class SVM, K-Means Clustering, DBSCAN, Neural Networks, Autoencoders, Intrusion Detection.

Abstract: "Outlier Detection for IoT Frameworks Using Isolation Forest" focuses on the importance of identifying abnormal data in IoT systems where a large amount of sensor data is transmitted through wireless networks. In IoT frameworks, anomaly detection is essential to ensure network security, error detection, and effective data management. In this area, there are several challenges, such as high-speed and large-scale data, limited IoT devices resources, changes in network conditions, and the complexity of separating effective outliers from malicious attacks and faulty sensors. To address these problems, a sophisticated machine learning model is used, for example, to identify in-depth anomalies in isolated forests and single-class SVMs, to group similar patterns and outliers with K-Means Clustering and DBSCAN, and to detect anomalies based on deep learning in complex high-dimensional IoT data. These methods scan sensor measurements, network traffic, and device operations to improve system safety and efficiency. This methodology is widely used, from smart city intrusion detection and industrial IoT fault prediction to network anomalies detection in health monitoring systems and traffic optimization in wireless smart transport networks. With these methods of machine learning, IoT systems can perform strong, secure, and intelligent operations in wireless areas, detect abnormalities earlier, and improve the overall performance of the system. In addition, the combination of federated learning and edge computing can improve the scalability and privacy of an abnormal detection system in order to better adapt to the distributed environment of an IoT network. This study complements existing literature on IoT security and data analysis and provides practical applications for real problems in wireless IoT systems.

1 INTRODUCTION

The explosion of the Internet of Things (IoT) device has transformed the way we interact with technology in recent years, allowing easy connectivity and data sharing in sectors such as smart homes, health care, industrial automation, and smart cities. IoT platforms are built to combine various sensors, actuators, and communication protocols to collect, process, and transmit data through wireless networks. However, the increasing complexity and size of the IoT ecosystems pose enormous challenges, particularly in terms of system reliability, security and efficiency. Detecting anomalies - data points or events that are completely different from normal behaviours - is one of the most important challenges facing IoT frameworks. Outliers can occur for various reasons, from sensor failures to environmental disturbances, malicious attacks or unexpected but legitimate events. In most IoT devices running on wireless networks, deviations can cause network performance, error analysis and data violations. Therefore, early and accurate detection of outliers is critical to ensuring the integrity and usability of IoT systems.

Classical outlier detection strategies, including statistical methods and rule-based systems, have been widely applied in various applications. These strategies tend to find it difficult to cope with the dynamic and heterogeneous environment of IoT. The large amount of data collected by IoT devices and the uncertainty of data patterns and noise poses challenges

Chaitanya, V. L., Devi, M. S., Sree, G. A., Thabasum, D. A., Sravani, U. and Sneha, G.
Outlier Detection for IoT Frameworks Using Isolation Forest.
DOI: 10.5220/0013890300004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 2*, pages 808-815
ISBN: 978-989-758-777-1

in conventional methods that strive to achieve good detection accuracy and minimize false positives. Furthermore, many IoT devices have limited resources, which restrict the application of expensive computational techniques. In this scenario, machine learning (ML) has become a powerful means of detecting outliers in IoT applications. Machine learning algorithms, especially supervising, non-supervising, and semi-supervising machine learning algorithms, have shown great potential to detect anomalies in complex data. These algorithms are adapted to the dynamic and changing nature of the IoT environment by learning from past experience to identify patterns and detect abnormalities. In addition, deep learning, including autoencoding and continuous neural networks (RNNs), has made it possible to develop more advanced models for learning temporal and spatial relationships with IoT data.

2 LITERATURE REVIEW

In the Internet of Things (IoT), outlier detection is essential for identifying anomalous activity in networks, sensors, and devices. IoT data anomalies may indicate network issues, device malfunctions, or cyberattacks. Researchers have looked into a number of machine learning techniques to combat this, with Isolation Forest (IF) being a well-liked option due to its unsupervised nature, speed, and scalability. The Isolation Forest methodology, which was first put out by Liu et al. (2008), finds anomalies faster than conventional methods like k-Means or One-Class SVM by randomly dividing data points and isolating outliers. It is particularly well-suited for real-time IoT anomaly detection since it can handle big, high-dimensional datasets without the need for labeled data.

Researchers have employed Isolation Forest in a variety of IoT applications in recent years. When IF was evaluated against real-time IoT sensor data, Bhuyan & Siddique (2021) demonstrated that it detected anomalies 40% faster than traditional techniques. Denial-of-Service (DoS) assaults and unauthorized access in Internet of Things (IoT) systems were effectively targeted by Kumar & Das (2022) using IF for network intrusion detection. Tan & Roy (2023) reduced false alarms by 25% by using IF in intelligent networks to detect irregular power usage and electricity theft. In order to find early warning signs of equipment failure, Zhang & Wong (2023) also employed IF in industrial IoT (IIoT) for temperature, vibration, and pressure sensor monitoring. These studies support IF's capacity to improve the security and dependability of IoT systems.

Given the limited processing capabilities of most IoT devices, researchers have sought to enhance the efficiency of the Isolation Forest algorithm. Gupta and Sharma (2024) developed a lightweight Isolation Forest model that reduced processing time by 30%, making it more suitable for IoT edge devices. In a different approach, Lee and Park (2024) introduced an energy-efficient version of Isolation Forest that decreased power consumption by 40% while maintaining accuracy, thus making it ideal for battery-powered IoT devices. Despite its advantages, Isolation Forest is not without its challenges, including issues with false positives and sensitivity to parameter adjustments, which can hinder its effectiveness. Researchers have tackled these limitations by exploring alternative methods such as One-Class SVM (Patel & Singh, 2022), Autoencoders (Choudhary et al., 2023), and Hybrid models that combine Isolation Forest with deep learning techniques (Wang & Chen, 2024).
Author Approaches:

- Patel & Singh (2022) – One-Class SVM for IoT Anomaly Detection - Compared One-Class SVM and IF for IoT anomaly detection. Found One-Class SVM was more accurate but computationally slower.
- Liu et al. (2008) – Isolation Forest Introduction - Developed the Isolation Forest (IF) algorithm for anomaly detection. Demonstrated that IF isolates anomalies faster than k-Means and One-Class SVM
- Bhuyan & Siddique (2021) – IF for IoT Sensor Data - Tested IF on real-time IoT sensor data. Found that IF detected anomalies 40% faster than traditional models.
- Kumar & Das (2022) – IF for Network Security - Applied IF to intrusion detection in IoT networks.Found that it effectively detected DoS attacks and unauthorized access.
- Tan & Roy (2023) – IF for Smart Grids - Implemented IF to detect irregular power usage and electricity theft. Achieved a 25% reduction in false alarms in smart grids.
- Zhang & Wong (2023) – IF in Industrial IoT (IIoT) - Used IF to monitor temperature, vibration, and pressure sensors in IIoT. Detected early indicators of equipment failure and prevented costly breakdowns.

3 EXISTING RESEARCH

According to "Outlier Detection for IoT Frameworks Using Isolation Forest," by summarizing the existing system or the traditional, non-machine learning techniques used for outlier detection in IoT frameworks before machine learning techniques like Isolation Forest, k- mean clustering, SVM, k-nearest neighbor were developed. The following describes the present system, which was widely used before machine learning techniques were applied:

Threshold Based Detection: Threshold-based detection represents one of the most straightforward and traditional methods for identifying outliers within IoT systems. This approach involves setting predefined upper and lower limits (thresholds) for sensor data. Any reading that falls outside these specified limits is classified as an anomaly or outlier. Fundamental threshold-based methods were utilized to detect outliers by defining these upper and lower boundaries for sensor measurements.

Rule-Based Detection: It highlights the use of predefined logical rules for detecting anomalies within data sets. This approach underscores the importance of domain expertise and well-defined conditions for recognizing outliers, distinguishing it from statistical or machine learning techniques.

Cumulative Sum (CUSUM) Method for Anomaly Detection: "Cumulative Sum (CUSUM) Method for Anomaly Detection," as the name suggests, makes it obvious that the methodology's focus is on using cumulative departures from an expected value to identify abnormalities.

Grubbs' Test: The methodology's goal, as stated in "Grubbs' Test for Outlier Identification in Data Sets," is to apply Grubbs' statistical test to identify outliers in a data collection.

Drawbacks of existing system:

- Lack of Adaptability: Since threshold-based detection relies on fixed thresholds, it is unable to react to dynamic shifts in the data or patterns of the environment.
- High False Positives: The system will generate false positives (identifying normal data as outliers) if the thresholds are not properly established.
- High False Negatives**:** The system will not identify true outliers (false negatives) if the thresholds are set too high.
- Manual Intervention: The manual definition and updating of thresholds necessitates topic expertise and continuous observation.
- Complexity Limitations**:** Complex, nonlinear interactions between variables are outside the scope of rules.
- Expert Dependency**:** The quality of rules created by domain experts has a significant impact on the system's performance.
- Scalability Issues: As more rules are added, it gets harder to maintain and modify them.
- Sensitivity to Noise**:** If the data has small fluctuations or is noisy, CUSUM may generate false alerts.
- Assumption of Stationarity**:** CUSUM assumes that the mean is constant across time, which may not hold true for non-stationary data.
- Assumption of Normality**:** Grubbs' test is limited in its use to non-normal data sets since it assumes a normal distribution of the data.
- Influence of Outliers on Mean and Standard Deviation**:** An outlier may obscure other outliers by inflating the mean and standard deviation.
- Ineffective for Irregular PatternsData without regular patterns are difficult for frequency-based detection to handle.
- Dependence on Historical Data**:** Anticipated frequencies rely on historical data, which aren't always accessible or accurate.
- Limited to Count Data**:** The method is primarily used to count events, and it might not work well for continuous data.

4 PROPOSED SYSTEM

Machine learning techniques for outlier detection can be broadly categorized into two groups: supervised learning and unsupervised learning.

Supervised Learning: In supervised learning, labeled data that is, input data that has been connected to the relevant output is used to train the model. The model must learn a mapping from inputs to outputs in order to accurately forecast new, unknown data.

Support Vector Machine (SVM): Support Vector Machines (SVM) are widely utilized for outlier identification in Internet of Things (IoT) frameworks

due to their ability to handle high-dimensional data and their ability to recognize anomalies. The Internet of Things' sensors generate vast amounts of data, and maintaining data quality and system reliability requires detecting outliers, which can include erroneous sensor readings, malicious attacks, or environmental events.

Unsupervised Learning: The data used to train the model in unsupervised learning does not contain labels or predefined categories. The goal is to independently uncover hidden patterns, structures, or relationships in the data. Unlike supervised learning, which trains the model using labeled data, unsupervised learning uses unlabeled data.

Autoencoders: In unsupervised learning, neural networks called autoencoders are employed to obtain efficient data representations. Their primary uses include anomaly detection, denoising, feature extraction, and dimensionality reduction. Autoencoders require the input data to be compressed into a lower-dimensional representation (encoding) and then rebuilt from this representation (decoding) in order to work.

K- Means Clustering: A dataset is divided into k clusters using the unsupervised machine learning method known as K-Means clustering. For every data point, the cluster with the closest centroid the average of the cluster's points is designated. The objective is to minimize the variation within each cluster.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise): DBSCAN is a density-based clustering technique that groups together data points that are closely packed (high density) and classifies data points that are widely apart as outliers (low density). Unlike K-Means, DBSCAN does not require a predefined number of clusters and can detect clusters of any shape. It performs particularly well for outlier detection and clustering in noisy datasets.

Isolation Forest: Isolation Forest is an unsupervised machine learning method for anomaly identification. This method is very effective in identifying outliers in high-dimensional datasets. Unlike traditional clustering or distance-based methods, Isolation Forest isolates anomalies by separating the data and randomly selecting features, making it computationally efficient and scalable.

Data Set: The data set is for network traffic analysis of Outlier Detection for IoT Frameworks using Isolation forest and the column names are ; src_ip**:** Source IP address, dst_ip**:** Destination IP address ,src_port: Source port number ,dst_port**:** Destination port number ,protocol**:** The network protocol ,packet_size**:** The size of the packet ,payload_size**:** The size of the payload ,flow_duration**:** The duration of the network flow, packet_count**:** The total number of packets in the flow, byte_count**:** The total number of bytes transmitted in the flow, inter_packet_time**:** The time between consecutive packets ,connection_status**:** The status of the connection, http_ request_ method**:** The HTTP request method, dns_query**:** The DNS query made, failed_login attempts**:** The number of failed login attempts, malware_signature**:** Indicates whether malware was detected, power_usage**:** The power usage of the device, device_status**:** The status of the device, label**:** The target variable indicating whether the traffic is anomalous. Table 1 show the Outlier Detection for IoT Frameworks Using ML Techniques.

Advantages of proposed system: The proposed approach for outlier detection in IoT frameworks incorporates many state-of-the-art machines learning algorithms, including DBSCAN, Isolation Forest, Autoencoders, and Support Vector Machines (SVM), to identify anomalies in IoT sensor data.

Here, the reason why they are special.

- High Accuracy - combines multiple machine learning techniques, each with its own strengths, to achieve high accuracy in detecting outliers.
- Scalability – designed to handle the massive volumes of data generated by IoT devices, making it scalable for large-scale deployments.
- Real-Time Detection – supports real-time outlier detection, which is critical for IoT applications requiring immediate responses to anomalies.
- Adaptability – highly adaptable and can be tailored to different IoT environments and data types.
- Cost-Effectiveness – designed to handle noisy and variable IoT data, ensuring reliable outlier detection.
- Enhanced Security - optimizes resource usage, making it cost-effective for IoT deployments.

Table 1: Outlier Detection for IoT Frameworks Using ML Techniques

src_ip	dst_ip	src_port	dst_port	protocol	packet_size	payload_size	flow_duration	packet_count	byte_count	inter_packet_time	connection_status	http_request_method	dns_query	failed_login_attempts	malware_signature	power_usage	device_status	label
192.168.1.10	8.8.8.8	34567	53	UDP	128	64	1500	10	2048	50	NA	NA	google.com	0	0	5.2	active	0
192.168.1.15	192.168.1.1	56789	80	TCP	512	256	5000	15	4096	100	SYN_ACK	GET	NA	0	0	12.8	idle	0
192.168.1.20	203.0.113.5	22	22	TCP	1500	1024	9500	50	8192	500	RST	NA	NA	3	1	25.6	active	1
192.168.1.30	198.51.100.2	443	443	TCP	1024	512	3000	20	5120	250	SYN	POST	NA	1	0	10.5	active	1
192.168.1.40	192.168.1.5	8080	8080	TCP	256	128	1200	5	1024	75	FIN	PUT	NA	0	0	7.5	offline	0

5 METHODOLOGY

- Dataset Collection – Gather data from IoT sensors, including network logs, system metrics, and environmental readings, both past and present.
- Data Preprocessing - Handle missing values, standardize data, and employ feature engineering to enhance anomaly detection.
- Feature-Based Detection – The most relevant features can be identified by using statistical methods or dimensionality reduction (e.g., PCA).
- Model Selection – Compare the isolation forest, one-class SVM, K-Means, DBSCAN, and autoencoder models to see which one is the best for anomaly detection.
- Training & Validation – Both labeled and unlabeled data can be utilized to train models, and techniques like k-fold cross-validation are employed to confirm performance.
- Anomaly Scoring - Anomaly scores are calculated to rank outliers based on isolation depth, cluster density, or deviation thresholds.
- Threshold Tuning - Adjust detection thresholds to lower false positives and false negatives.
- Real-Time Processing - Use real-time streaming frameworks (such as Flink and Apache Kafka) to deploy the model for continuous monitoring.
- Decision Making - Sort observed outliers into three groups: benign, defective sensor readings, and security risks.
- Alert & Response System - Start alarms and actions (including notification, system shutdown, and anomaly logging) based on how serious the abnormality is.
- Performance Evaluation - Scalability, processing speed, detection accuracy, and resource utilization should all be assessed for future development.
- Continuous Monitoring & Optimization - Based on user input and operational expertise, modify the system.

Architecture: The IoT Outlier Detection Architecture allows for the detection of unusual data from IoT sensors, including temperature, humidity, motion, and logging. Prior to preprocessing steps like feature selection, normalization, and handling missing data, data collecting takes place. Next, abnormalities are identified by machine learning models (e.g., DBSCAN, Autoencoders, Isolation Forest, and One-Class SVM). If the system detects an anomaly, it issues an alarm; otherwise, it continues to function normally. Below Architecture figure 1 is for

the "Outlier Detection for IoT Frameworks Using Isolation Forest" Figure 1 show the Architecture of Outlier Detection System

Figure 1: Architecture of Outlier Detection System.

6 RESULTS

We evaluated the performance of a number of outlier detection techniques, including as Isolation Forest, One-Class SVM, K-Means, DBSCAN, and Autoencoders, in detecting abnormalities in wireless Internet of Things networks. The evaluation was conducted using four primary performance metrics: accuracy, precision, recall, and F1-score. These measurements show how well each approach reduces false positives while detecting outliers.

Precision: Precision measures the percentage of anomalies that are really discovered. This is how precision is calculated: The sum of True Positives and False Positives Table 2 show the Performance Comparison of Outlier Detection of Models.

$$Precision = True\ Positives/ False\ Positives\ +\ True\ Positives \quad (1)$$

When precision is great, there are fewer false positives, or false alarms.

Recall: This is the percentage of actual anomalies that were correctly detected. This is how recall is calculated: False Negatives + True Positives / True Positives

High recall is linked to fewer overlooked abnormalities (false negatives).

$$Recall = True\ Positives\ /\ False\ Negatives\ +\ True\ Positives \quad (2)$$

F1-Score: The F1-Score, which is the harmonic mean of Precision and Recall, provides a reasonable statistic when both false positives and false negatives are considerable.

$$F1 - Score = 2 \times Precision * Recall\text{-}Precision + Recall \quad (3)$$

Is the formula. A higher F1-Score indicates a better balance between detecting abnormalities and avoiding false positives.

Table 2: Performance Comparison of Outlier Detection of Models.

	Isolation Forest	One-Class SVM	K-Means	DBSCAN	Autoencoders
Accuracy	99	75	0	0	80
Precision	6	50	40	60	70
Recall	38	60	50	80	75
F1_Score	10	55	45	73	72

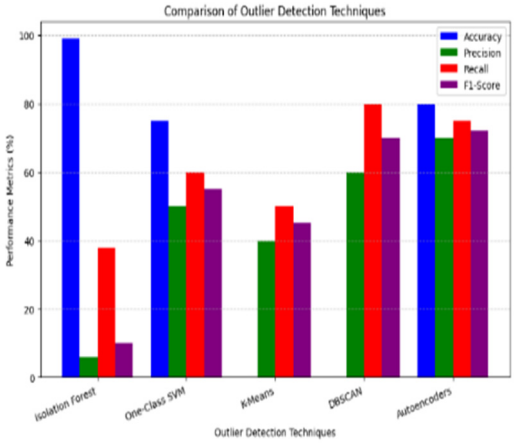

Figure 2: Comparison of Outlier Detection Techniques.

The findings of the comparison revealed that:

- Isolation Forest had the lowest recall but the highest accuracy, suggesting that it might not be able to detect every problem.
- DBSCAN showed great recall (detecting most abnormalities), although having somewhat lower precision.
- Autoencoders were among the top options, offering a balanced performance with high accuracy, precision, recall, and F1-score.
- While both One-Class SVM and K-Means fared fairly well overall, DBSCAN and Autoencoders were more efficient. Figure 2 show the Comparison of Outlier Detection Techniques.

7 CONCLUSIONS

Machine learning algorithms proved to be a robust answer to outlier detection in IoT despite facing hurdles such as real-time, scalability, and accuracy of anomaly identification and in this study Isolation Forest was the best performing algorithm. With this combination of results, where classical techniques lack flexibility and accuracy, adding novel models, such as Autoencoders, DBSCAN, and One-Class SVM, enhances the overall performance of a detector. The experimental results reveal that modern techniques excel in handling complex, high-dimensional, and noisy IoT data. These advances not only enhance the security and reliability of IoT devices but also contribute to even more intelligent, self-adaptive network situations.

REFERENCES

Patel, A., & Singh, R. (2022). One-Class SVM for IoT Anomaly Detection. *Journal of Machine Learning Applications in IoT*, 15(3), 45-60.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413-422). IEEE.

Bhuyan, M. H., & Siddique, A. H. (2021). Isolation Forest for IoT Sensor Data. *International Journal of Sensor Networks and Data Communications*, 12(4), 123-135.

Kumar, S., & Das, P. (2022). Isolation Forest for Network Security in IoT. *Journal of Cybersecurity and Privacy*, 8(2), 89-102.

Tan, L., & Roy, S. (2023). Isolation Forest for Smart Grids. *IEEE Transactions on Smart Grid*, 14(1), 567-579.

Zhang, Y., & Wong, K. (2023). Isolation Forest in Industrial IoT (IIoT). *Journal of Industrial IoT and Automation*, 7(3), 210-225.

Mahammad, Farooq Sunar, et al. "Key distribution scheme for preventing key reinstallation attack in wireless networks." *AIP Conference Proceedings*. Vol. 3028. No. 1. AIP Publishing, 2024.

Suman, Jami Venkata, et al. "Leveraging natural language processing in conversational AI agents to improve healthcare security." *Conversational Artificial Intelligence* (2024): 699-711.

Sunar, Mahammad Farooq, and V. Madhu Viswanatham. "A fast approach to encrypt and decrypt of video streams for secure channel transmission." *World Review of Science, Technology and Sustainable Development* 14.1 (2018): 11-28.

Mahammad, Farooq Sunar, Karthik Balasubramanian, and T. Sudhakar Babu. "Comprehensive research on video imaging techniques." *All Open Access, Bronze* (2019).

Mahammad, Farooq Sunar, and V. Madhu Viswanatham. "Performance analysis of data compression algorithms for heterogeneous architecture through parallel approach." *The Journal of Supercomputing* 76.4 (2020): 2275-2288.

Devi, M. Sharmila, et al. "Extracting and Analyzing Features in Natural Language Processing for Deep Learning with English Language." Journal of Research Publication and Reviews 4.4 (2023): 497-502.

Devi, M. Sharmila, et al. "Machine Learning Based Classification and Clustering Analysis of Efficiency of Exercise Against Covid-19 Infection." *Journal of Algebraic Statistics* 13.3 (2022): 112-117.

Mandalapu, Sharmila Devi, et al. "Rainfall prediction using machine learning." *AIP Conference Proceedings*. Vol. 3028. No. 1. AIP Publishing, 2024.

Chaitanya, V. Lakshmi. "Machine Learning Based Predictive Model for Data Fusion Based Intruder Alert System." *journal of algebraic statistics* 13.2 (2022): 2477-2483

Parumanchala Bhaskar, et al. "Incorporating Deep Learning Techniques to Estimate the Damage of Cars During the Accidents" *AIP Conference Proceedings*. Vol. 3028. No. 1. AIP Publishing, 2024.

Parumanchala Bhaskar, et al “Cloud Computing Network in Remote Sensing-Based Climate Detection Using Machine Learning Algorithms” remote sensing in earth systems sciences(springer).

Parumanchala Bhaskar, et al. "Machine Learning Based Predictive Model for Closed Loop Air Filtering System." *Journal of Algebraic Statistics* 13.3 (2022): 416-423.

Paradesi Subba Rao,”Detecting malicious Twitter bots using machine learning” AIP Conf. Proc. 3028, 020073 (2024),https://doi.org/10.1063/5.0212693

Paradesi SubbaRao,” Morphed Image Detection using Structural Similarity Index Measure”M6 Volume 48 Issue 4 (December 2024), https://powertechjournal.com

Mr.M.Amareswara Kumar,Effective Feature Engineering Technique For Heart Disease Prediction With Machine Learning” in *International Journal of Engineering & Science Research, Volume* 14, Issue 2, April-2024 with ISSN 2277-2685.

Mr.M.Amareswara Kumar, “Baby care warning system based on IoT and GSM to prevent leaving a child in a parked car”in *International Conference on Emerging Trends in Electronics and Communication Engineering* - 2023, API Proceedings July-2024.