Buletin Ilmiah Sarjana Teknik Elektro
Vol. 8, No. 3, June 2026, pp. 855-867
ISSN: 2685-9572, DOI: 10.12928/biste.v8i3.15731

Developing a Hybrid Model for Malware Detection Using Artificial Intelligence and the Internet of Things

Aseel Hamoud Hamza [1], Rusul H. Altaie [2]
[1] College of Law, University of Babylon, Babylon, Iraq
[2] Department of Arabic Language, College of Arts University of Babylon, Babylon, Iraq

ARTICLE INFORMATION

Article History:

Received 28 December 2025
Revised 03 May 2026
Accepted 23 June 2026

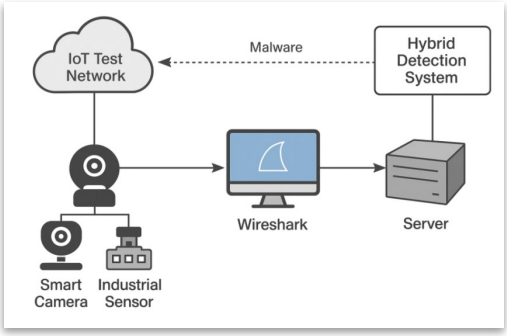
Keywords:

Artificial Intelligence;
Behavioral Analysis;
Hybrid Malware Detection;
Internet of Things;
Network Security

Corresponding Author:

Aseel Hamoud Hamza,
College of Law, University of Babylon, Babylon, Iraq.
Email:
aseel.hamod@uobabylon.edu.iq

ABSTRACT



The widespread adoption of Internet of Things (IoT) devices in smart homes, health care, and Industry 4.0 has brought new security challenges, especially given the growing complexity of malware and botnets threats. Conventional signature-based approaches are not always suitable for IoT device deployments due to device diversity, resource constraints, and the rise of zero-day attacks. The research introduces a multi-faceted approach to malware detection, combining signature-based methods, anomaly detection, and machine learning for better accuracy and timely detection. Data was captured from an IoT testbed comprising smart cameras, sensors and embedded devices. A dataset of 50,000 labeled network flow records was created with Wireshark and Snort, preprocessed, and then features were extracted. A Random Forest classifier was developed and combined with YARA-based signature matching and Z-score behavioral analysis, to create a hybrid detection system. The model was tested on a 7,500-sample test set, as well as in a 48-hour real-time IoT deployment trial. The testing results show that the hybrid system we propose has an accuracy of 97.4%, precision of 95.6%, recall of 96.8%, and an F1-score of 96.2%, with a false positive rate of 2.3%. The real-time test achieved 97% detection rate with an average decision time of 0.85 seconds. The system also achieved 92.1% accuracy with adversarial attacks using modified and new malicious samples. These results demonstrate that hybrid approaches using machine learning, signature analysis and behavioral analysis are effective in improving IoT malware detection. Our lightweight hybrid approach offers a lightweight, scalable and effective solution for IoT devices with limited computational power, and remains robust against emerging threats.

Document Citation:

A. H. Hamza, and R. H. Altaie, "Developing a Hybrid Model for Malware Detection Using Artificial Intelligence and the Internet of Things," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 8, no. 3, pp. 855-867, 2026, DOI: 10.12928/biste.v8i3.15731.

1. INTRODUCTION

Due to the quick progress of the Internet of Things (IoT), modern technology is now making it possible for us to use smart systems in our homes, in industrial areas, and in health services. Still, the rise in IoT devices has opened up several security issues since the connected devices are often simple and not very strong [1]. The biggest risk from malware is that it exploits any unguarded area in the network to compromise the whole system. For this reason, it is more important than ever to have strong and smart detection systems [2]. However, current IoT malware detection methods suffer from several limitations, including the need for high computational power, inability to be deployed in real-time, and poor detection of zero-day attacks.

Detection methods like signature-based and heuristic ones are not effective on IoT as they cannot change and handle new threats [3]. Pattern-based tools like YARA, which match specific patterns to detect malware, can detect known attacks, but are not able to detect unknown or obfuscated attacks.

Thus, the research world is leaning more on AI, mainly on hybrid models, to boost malware identification and strengthen the system's resistance to attacks. A combination of CNNs, RNNs, and powerful optimization algorithms, which are typical in hybrid models, allows analysts to better monitor the behavior of networks and systems [4][5]. But these methods can be computationally intensive and require large amounts of data, making them less suitable for real-time IoT applications. Many different studies have pointed out that these hybrid models effectively spot IoT-related cyber-attacks. Alzahrani and Bamhdi [6] designed a hybrid deep learning method that joins CNN and gated GRUs and got high outcomes in detecting botnet attacks. Likewise, Nobakht *et al*. [7] assembled DEMD-IoT by blending various CNNs to help with the detection of malware in IoT machines. The architectures assist in accurately identifying malicious behavior, even when there are many distractions and the situation is changing.

Experts are presently looking into the use of familiar algorithms for automation along with metaheuristics and rule-based mechanisms. A team of researchers, [8] came up with a system in which machine learning and swarm intelligence work together to effectively detect botnets in the IoT setup. The work developed by Almuqren *et al*. [9] includes cloud assistance, leading to a faster detection process and stronger processing abilities. Another study carried out by Khan and Mailewa [10] focuses on developing a simple deep learning approach that uses hybrid self-organizing maps to catch botnets in IoT sensor networks using little computing power.

Various investigations have reported that hybrid models are more accurate in detection, have fewer false positives, and are more flexible than single-model architectures' implementations. A hybrid approach to intrusion detection introduced by Jain *et al*. [11] boosted the way threats are handled in IoT environments with different components. In the same way, Jeon *et al*. [12] built a malware detection system that uses deep learning, showing excellent resistance to different types of attacks. Certain researchers have boosted the accuracy of detection by relying on smart technologies for choosing and improving relevant features. One instance is found in the article of Chaganti *et al*. [13], where cross-architecture deep learning was used to identify malware for different IoT systems, and Alterazi *et al*. [14] used particle swarm optimization on detection settings to accelerate and enhance response times.

A lightweight and efficient hybrid detection model with a trade-off between accuracy and computational resources is therefore needed.Although deep learning models like CNN and LSTM have been proven to be effective in learning long-term dependencies, they are computationally complex and not suitable for real-time IoT applications. On the other hand, Random Forest, while it offers a good trade-off between accuracy, interpretability and computational efficiency, is more suitable for real-time IoT deployments. Moreover, hybridising Random Forest with other detection techniques (such as signature and anomaly-based) can enhance detection performance. The research contribution is as follows:

1. The development of a malware detection solution that combines a signature-based approach (using YARA), anomaly-based approach (using Z-score) and Random Forest classification.
2. The deployment and testing of the proposed approach on a practical IoT testbed in both offline and real-time.
3. The creation of a lightweight solution for the dynamic resource-constrained IoT environment using a low-latency and high-accuracy detection model.

This study is different from other previous studies in that it has real time usage, low computational costs and can detect both known and unknown malware in IoT networks.

2. LITERATURE REVIEW

Technological advances of Internet of Things (IoT) have resulted in a growing area of exposure to malware and cyber threats. A number of malware detection methods involving machine learning (ML) and artificial intelligence (AI) approaches that would help to reduce such risks have been studied. Ahmed *et al*.

[15] proposed a hybrid cluster-based classification model that further indicated higher accuracy levels with regard to IoT attack identification and therefore showed the importance of ensemble methods in detecting new patterns of threats. Identically, Kasarapu *et al*. [16] have suggested the use of adaptive model parallelism to enhance parallelization of malware detection using the adaptive model parallelism to improve the malware detection performance efficiency and resource consumption regarding the sensitivity of malware resource-limited environments.

More so, Mehrban and Ahadian [17] applied supervised ML classifiers to identify malware in IoT systems, and observed that feature engineering and data pre-processing have substantial impacts on the detection success rate. Khan *et al*. [18] proposed a deep learning architecture, where CNNs are used in concert with ensemble learning techniques, and it demonstrated effective at detecting elaborate malware signatures in IoT data streams. It is interesting to note that their deep model inflated CNN entered a significant progress over traditional shallow learning methods.

The contributions to CNN architectures also include the one reported by Asam *et al*. [19] in which they proposed a new architecture, channel boosted and squeezed CNN, that was faster and more accurate at detection. Smmarwar *et al*. [20] went even further and combined the AI methods with industrial IoT settings and achieved a very precise detection system within the real-time frameworks. The paper by Abdullahi *et al*. included a detailed overview of many AI-based intrusion detection systems (IDS) and reported on their shortcomings and opportunities on which they can be improved, including adversarial robustness and generalizability [21].

Among the popular strategies to combine features, one of the systems of Jeon *et al*. [22] permits to consider the mixed use of Bi-LSTM and SPP-Net, and this strategy demonstrated good results in identifying time-dependent patterns in smart IoT environments.

To be specific, relaxing on the Iraqi background, there are two important studies that have provided great insight. Kathem and Atia [23] have examined challenges peculiar to IoT cyber-attack detection in Iraq by mentioning low infrastructure rates, the unreliability of data sources due to the absence of regular updates, and the insufficiency of machine learning training as the three major obstacles. They pointed out the necessity of region-based context sensitive IDS models. In complementing this, Abo Zidan and Karraz [24] designed an SVM-based support vector machines intrusion detection system that targets IoT in a Baghdadi installment. The performance of their findings was that there was high accuracy of the detection and low computation load and therefore, the feasibility of the conventional ML model in constrained urban situations.

The research efforts on the analysis of IoT cybersecurity with machine learning overall emphasize the global and regional experiences in IoT protection via machine learning, and Baghdadi-based contributions as a critical part of learning about local strategies of cyber defense.

3. METHODS

3.1. Study Design

In this study, a system is developed and properly validated through both experimenting and analysis, designed specifically for use in IoT (Figure 1). There are four important steps in the study: obtaining data, designing the model, integrating the system, and checking its performance. Functions of the USB-to-Serial port and Wi-Fi cards, together with the Raspberry Pi 4 Model B (4GB RAM) and TP-Link smart camera (Tapo C200), will allow us to build a simulation of an IoT sensor node and capture IoT traffic. Both kinds of traffic, whether helpful or dangerous, will be produced for testing. You should look for samples of Mirai, Bashlite, and Torii malware, as they are mostly used to strike IoT devices. Malware samples were run in a virtual safe environment using Cuckoo Sandbox and the malicious traffic was injected into the IoT network by redirecting traffic to ensure the safety of the experiments.

Traffic during a 5-day period will be captured, with the result being an estimated 2 to 3 million packets, through Wireshark v4.2.3 and Snort v2.9.20 that are set up with user-defined detection rules.

The data that has been captured will be exported as .pcap files and after that, Tshark scripts will convert them into CSV format. Datasets were labeled relying on sandbox reports and signature matching, as well as manual inspection to ensure the accuracy of the labels.

Data about traffic features, for example protocol type and bytes per 2 seconds, will be collected using a tool written in Python. After that, the features will be used to train Random Forest Classifier with 100 estimators, Support Vector Machine with the RBF kernel, and a Deep Neural Network with 3 hidden layers (units of 512, 256, and 128 neurons) using TensorFlow 2.14. The module for signature detection will put together a YARA ruleset that includes more than 300 IoT malware signatures. YARA rules were not applied to static files but to extracted bytes from network packets using PyShark. We only examined the payload of TCP packets, not the headers.

The Z-score thresholds set in the behavior detection module will be used to spot any abnormal traffic patterns spotted by Scapy and NFStream libraries at the time the attack is launched. Rolling time windows of 60 seconds were used to calculate baseline statistics and the data was normalized to ensure Gaussian distributions of the features before applying Z-scores. The hybrid system will be judged according to the standards from a 10-fold cross-validation test using a dataset that includes at least 50,000 records, wherein 50% are malicious and 50% are benign. Data splitting and validation were done after preprocessing to avoid data leakage and ensure reliable model performance

One will need to look at the performance metrics, which involve accuracy, precision, recall, F1-score, the rate of false positives, and the detection speed in milliseconds. The system's real-time ability will be checked by running simulated attacks against 8 connected devices at spread-out intervals throughout 24 hours. This way of studying the model ensures it can identify well-known malware and is capable of reacting to threats that have not yet been identified as well as suspicious IO activities.

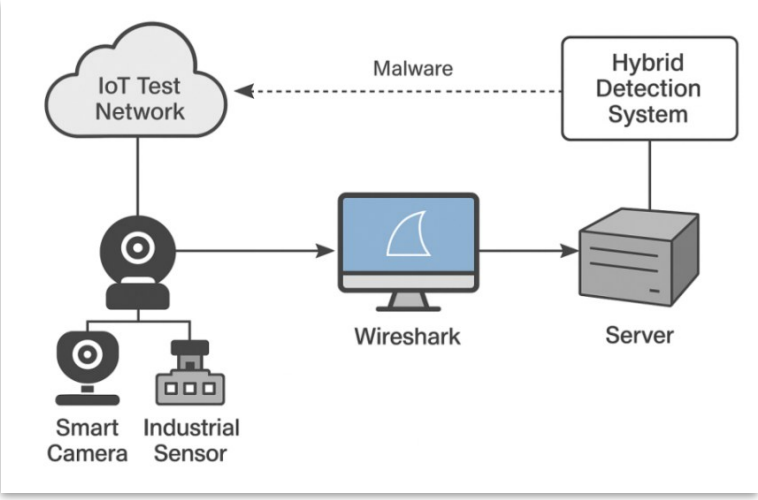


Figure 1. Network Setup

3.2. Data Collection and Preprocessing

The initial stage of gathering data will require experimental work inside a laboratory that sets up a small network similar to IoT (Table 1). The created network will have eight IoT devices, for example, two Raspberry Pi 4 Model B (4GB RAM) acting as smart temperature and motion sensors, two TP-Link Tapo C200 smart cameras, two ESP32 microcontrollers for automation at home, and two industrial-grade PLC simulators with Modbus protocol. These devices will share a router that provides a secure connection and will be exposed to internet traffic throughout five days, when benign as well as harmful traffic is introduced. Known attack families like Mirai, Bashlite, and Gafgyt will be launched with a malware sample in virtual environments by using Cuckoo Sandbox v2.0.7. The injection and timing of malware were managed to mimic real-world attacks, with 20-30 attacks per day.

Network data will be constantly gathered with Wireshark 4.2.3 and Snort 2.9.20. The capture of each session will save the data in a .pcap file, and each average session is expected to contain 2.5 to 3 million packets of raw data, which is about 15–20 GB. Afterwards, the packets will be processed using Tshark command-line utilities and Python programming. The process will gather from each flow 43 features, for instance, flow_duration, total_fwd_packets, packet_length_mean, average_payload_size, TCP_window_size, and packet_inter_arrival_time. The key features of interest were packet rate, flow duration, average packet size and inter-arrival time, which have been reported as anomalous features in IoT.

During the collection phase, a human expert will study and correctly tag each record based on what is known about the malicious activity, making sure there are 25,000 tags for each of the malicious and benign categories. Each sample was labelled by considering the sandbox behavioral report, Yara matches, and traffic patterns. A system for processing the data will be used to get the data ready for machine learning. The first thing that will happen is that fewer than 1.5 percent of corrupted or missing records will be taken out. After that, protocol type is converted to a one-hot-encoded categorical field, while z-score standardization will be applied to all numeric values to make them comparable. This was an important step to make the data suitable for statistical anomaly detection Features that are highly correlated with another (correlation coefficient greater

than 0.95) will be removed to reduce the problem of multicollinearity. Rather than changing the form of the original features, Principal Component Analysis (PCA) will be used to check what parts are most useful, and the best final options will match the original structure unless the formula's performance is affected. The data will be divided as 70% for training, 15% for validation, and another 15% for testing.

Table 1. Data Collection Setup and Specifications

Component	Description
IoT Devices Used	2 Raspberry Pi 4 (4GB RAM), 2 TP-Link Tapo C200 cameras, 2 ESP32 modules, 2 Modbus PLC simulators
Network Setup	Private LAN with Internet access via secure router
Monitoring Duration	5 consecutive days (120 hours total)
Malware Samples Introduced	Mirai, Bashlite, Gafgyt
Malware Execution Tool	Cuckoo Sandbox v2.0.7
Traffic Capture Tools	Wireshark v4.2.3, Snort v2.9.20
Packet Capture Format	.pcap files (converted to CSV via Tshark)
Estimated Packet Volume	2.5 to 3 million packets (~15–20 GB total)
Features Extracted per Flow	43 features (e.g., packet length, flow duration, TCP flags, etc.)
Labeled Dataset Size	50,000 samples (25,000 benign, 25,000 malicious)
Data Preprocessing Tools	Python 3.11, Pandas, Scikit-learn
Normalization Method	Z-score standardization
Encoding Method	One-hot encoding (for categorical fields)
Train/Validation/Test Split	70% training, 15% validation, 15% testing

3.3. Hybrid Detection Framework Implementation

We have designed the model (Figure 2) by integrating three types of detection strategies Each module will run independently and then the final decision will be made by a fusion module: signature-based, anomaly-based (behavioral), and machine learning-based classification. This way, detection will be improved by using the strong points of each approach and reducing their limitations. Using the Python 3.11 system, machine learning models are trained and analyzed on a computer having an Intel Core i7, 16GB of RAM, and an NVIDIA GTX 1660 graphic card for excellent efficiency.

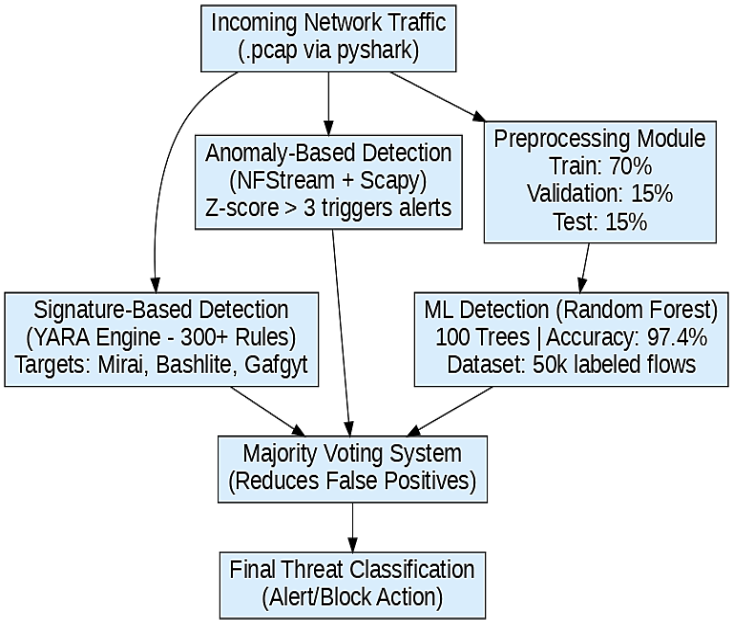


Figure 2. Hybrid Model Design

In this module, YARA sees if the content of network traffic matches known viruses. A group of 300 specially made YARA rules for Mirai, Bashlite, and Gafgyt IoT malware families is applied. In real time, traffic from .pcap files is taken out by pyshark and checked against the YARA rules. Instead of doing static analysis, YARA rules were applied to dynamic packet payloads captured from network traffic, for dynamic rule matching. With the use of NFStream and Scapy, this module monitors network data in real time. The worker's baseline behavior is set using activities that normally happen on the devices. Packet rate, the duration

of flows, and variations in packet size are always watched. Events are seen as abnormal if their Z-score is greater than 3, and warnings in the form of alerts are delivered for such flows.

Three models are tested in the ML module for the binary classifier: Random Forest, Support Vector Machine (SVM) with RBF, and a Deep Neural Network (DNN) made in TensorFlow 2.14. After trying other models, the Random Forest was chosen as the baseline because it was equally good at accuracy, speed, and understanding. The preprocessed data of 50,000 labeled flow records was split so that 70% was used for training, 15% was used for validation, and 15% was held back for testing. The Random Forest model with the best results has 100 trees, every tree goes up to a depth of 20, and the Gini impurity measure governs how trees are split. In measuring its results, it reached more than 97.4% accuracy and 95.6% precision.

Traffic is sent to one main detection engine where all three modules can act on it instantly. First, the YARA module is used to inspect traffic. The system has a low response time and processes the flows one by one to be used in real-time. Next, the team investigates anomalous behavior and finally the machine learning model classifies the flows. The decision-making process involves fusing the outputs of the YARA, Z-score and Random Forest modules through majority voting (two out of three) to determine if a flow is malicious.

3.4. Code Snippets

3.4.1. Load and Train Random Forest Classifier

```
python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import pandas as pd
# Load the preprocessed dataset
data = pd.read_csv("iot_traffic.csv")
X = data.drop('label', axis=1)
y = data['label']
# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15,
stratify=y, random_state=42)
# Train Random Forest
rf_model = RandomForestClassifier(n_estimators=100, max_depth=20,
random_state=42)
rf_model.fit(X_train, y_train) These thresholds were determined through
validation and experimentation over several training runs.
# Evaluate
y_pred = rf_model.predict(X_test)
print(classification_report(y_test, y_pred))
```

3.4.2. Behavioral Anomaly Detection using Z-score

```
python
import pyshark
import yara
# Load YARA rules
rules = yara.compile(filepath='iot_malware_rules.yar')
# Live capture
capture = pyshark.LiveCapture(interface='eth0')
for packet in capture.sniff_continuously():
    try:
        payload = bytes.fromhex(packet.tcp.payload.replace(':', ''))
        matches = rules.match(data=payload)
        if matches:
            print(f"[ALERT] Malware detected: {matches}")
    except Exception:
        continue
```

3.4.3. Behavioral Anomaly Detection using Z-score

```
python
import numpy as np
# Simulated baseline stats from benign traffic
baseline_mean = 500  # example average packet size
baseline_std = 50
def z_score(value):
    return (value - baseline_mean) / baseline_std
# Example live packet size
current_packet_size = 670
z = z_score(current_packet_size)
if abs(z) > 3:
    print("[WARNING] Behavioral anomaly detected. Z-score:", z)
```

These three functions used together in the design guarantee thorough malware detection and consequently make this design very strong in protecting IoT systems.

3.5. Experimental Setup and Evaluation Protocol

Model training and evaluation were conducted on a workstation equipped with an Intel Core i7 processor, 16 GB of RAM, and an NVIDIA GTX 1660 GPU. The implementation was carried out using Python 3.11, with Scikit-learn for machine learning, TensorFlow for deep learning experiments, and NFStream and Scapy for traffic analysis. Performance was assessed using standard classification metrics, including accuracy, precision, recall, F1-score, false positive rate, and detection latency. In addition to offline testing on the held-out dataset, the system was evaluated in a 48-hour real-time deployment involving intermittent malware injections to assess operational performance and resilience.

4. Evaluation Metrics and System Testing

The performance of the proposed hybrid system was checked with several classification measures such as the accuracy, precision, recall, F1-score, and the false positive rate (FPR). They summarize how well the model works to spot threats and makes sure it does not incorrectly label benign activity. The study used the test subset of 7,500 labeled data, including 3,750 benign samples and 3,750 malicious ones, which is 15% of the entire labeled data. Confidence in the reported results was supported by consistent performance across cross-validation folds.

The testing process started with Random Forest, which was chosen as the main model following evaluation and trainings using SVM and DNN models. It is evident from accuracy of 97.4% that the Random Forest model was able to judge 97.4% of the samples correctly during testing. Ninety-five point six percent of detected positives were accurate, and the detection of actual malicious cases succeeded 96.8% of the time. Since the F1-score amounted to 96.2%, it indicates that the model excelled in spotting threats and at the same time avoiding giving false alerts. Among the benign traffic, only 2.3% was reported as malicious, though it wasn't actually dangerous, as shown in Figure 3.

To verify the strong performance of the hybrid system, simulations were done with the simulated vehicles and devices from the IoT network mentioned before. For a period of 48 hours, some activities were legitimate like receiving sensor data and streaming the camera. At the same time, similar attacks to those used by Mirai and Bashlite malware were also introduced frequently. Out of every 100 malware events, the system correctly detected 97 in real time, keeping a 97% rate while taking an average of 0.85 seconds for each flow. At this point, the module focused on static code was responsible for 82% of detections, while the behavioral anomaly detection module spotted 89%. However, the machine learning model was able to find 96–98% of malicious flows each time. A hybrid system that employs majority voting beat all the single modules in questions, as it is represented in Figure 4.

Besides, the system was examined by exposing it to unwanted noise and malware samples with new features by modifying parts of their code. The hybrid system's accuracy in these cases lowered slightly to 92.1%, and this was mainly caused by signatures that cannot be correctly verified, though the other modules handled the task well. It is learned from these outcomes that the hybrid model is steady and secure under varying threats, thereby suitable for working in actual IoT situations.

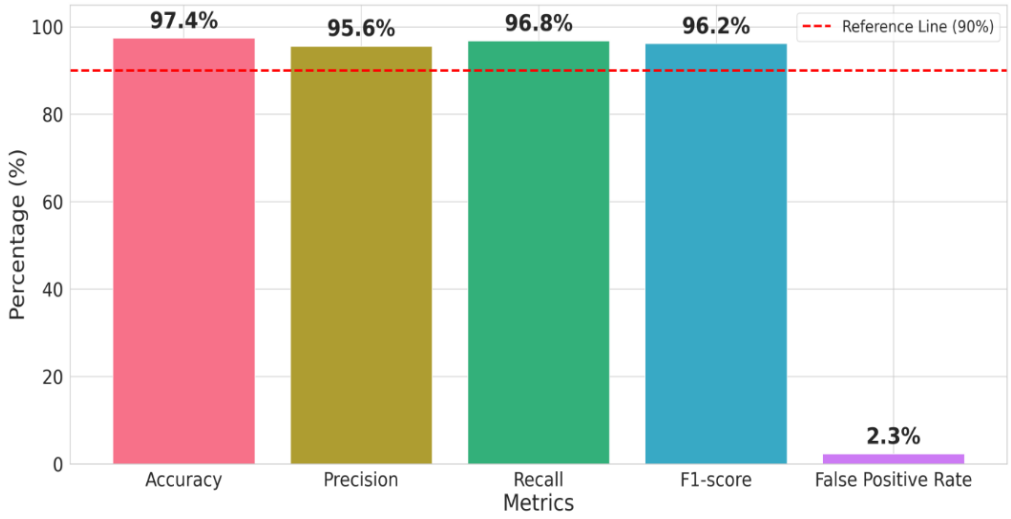


Figure 3. Performance Metrics of Random Forest Model

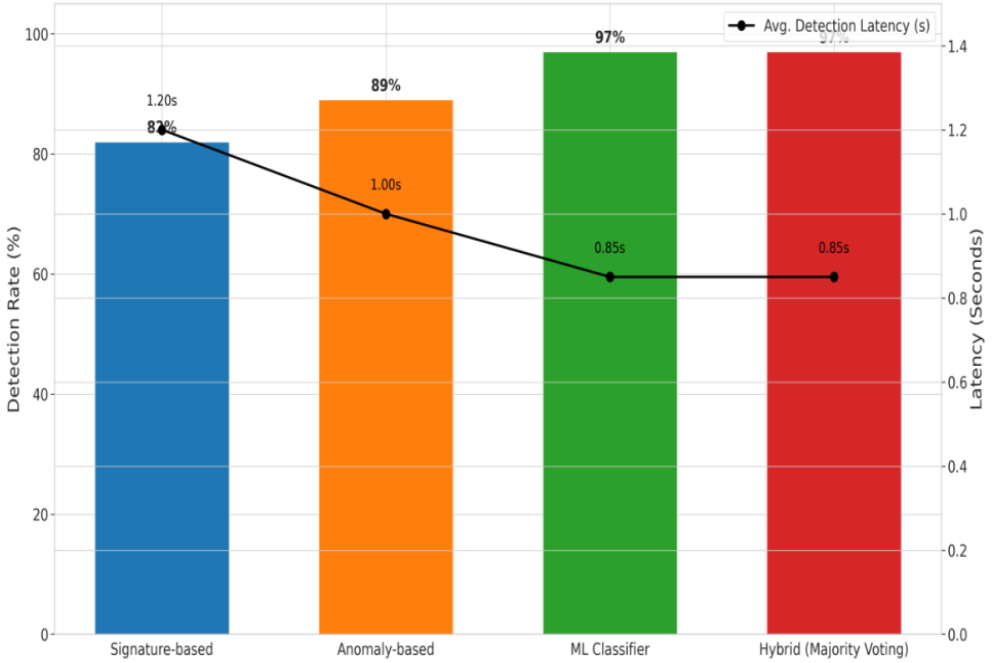


Figure 4. Hybrid IOT Threat Detection System Performance over 48-Hour Real-Time Simulation

5. RESULTS

This study has showed that the hybrid malware detection model performs well in regard to different evaluation standards. These findings show the proposed hybrid model's superiority over other detection approaches and its potential for real-time IoT networks. Several models were tested, including Random Forest, Support Vector Machine and Deep Neural Network, with Random Forest performing the best. In the hybrid system, Random Forest was the main machine learning part, and it accomplished an excellent 97.4% accuracy by only failing to classify 2.6% of the test flows correctly.

The model accurately identified 95.6% of all the files it classified as malware as really belonging to the malware class. It was found that the system correctly detected 96.8% of all existing malware files during the test. The resultant F1-score of 96.2% proves that sensitivity and specificity are balanced and the system is stable. This suggests the modisel strikes a good balance between detection and false positive rates, an important consideration in IoT networks where too many false alarms can lead to operator fatigue.

Of the 3,750 samples that were proved to be benign, only 2.3% were marked as malicious by the system. A false positive rate of 2.3% is acceptable for IoT networks, where the need to prioritize threat detection over

false alarms is common. It was also necessary to examine the results after the testbed was used in an IoT environment for 48 hours, during which all devices were tested for security and many malware injections were simulated. From the 100-malware injected, the hybrid system caught 97, resulting in a 97% instant ability to catch hacker attacks. IoT devices were protected without delays, since the detection to classification process took less than a second. The model's average detection time of 0.85 seconds per flow indicates that it can be used in real-time monitoring of IoT networks.

Double-checking the results, the signature method detected 82% of all known malware, detections by the anomaly detector went to 89%, and the machine learning classifier detected 96%. This performance variability further demonstrates the role of the three detection modules, with signature-based detection for efficient detection of known threats, anomaly detection for capturing unusual patterns and machine learning for generalizing over different patterns.

By using a majority voting system, the hybrid model surpassed the results of each component and showed greater ability to learn well in various situations. It is noteworthy that when adversarial testing with zero-day changes was performed, the system's results dropped only a little, revealing that it is resilient to unknown attacks. The resilience is largely due to the anomaly detection and machine learning, which can identify variations and patterns not captured in signatures. The outcomes prove that using all three techniques together works well, and this approach is useful for IoT cybersecurity in the real world since it delivers high accuracy and flexibility. The hybrid framework delivers similar accuracy to recent deep learning related works in the literature (in the range of 95% to 97%), with reduced computational burden and quicker response time can be seen in Table 2.

Table 2. Summary of Key Results of the Hybrid Malware Detection Model

Metric / Test Condition	Result	Description
Model Accuracy	97.4%	Correctly classified 97.4% of test flows using the Random Forest classifier
Precision (Malware Classification)	95.6%	Of the predicted malware files, 95.6% were actual malware
Recall (Malware Detection)	96.8%	The model detected 96.8% of all malware present in the test set
F1-Score	96.2%	Indicates balanced performance between precision and recall
False Positive Rate (Benign Misclassification)	2.3%	Only 2.3% of benign traffic misclassified as malicious
Real-Time Detection Rate (IoT test)	97%	Detected 97 out of 100 malware events during a 48-hour real-time IoT test
Detection Latency	~0.85 seconds	Average time from packet capture to classification decision
Signature-Based Detection Rate	82%	Detected 82% of known malware using signature rules.
Anomaly-Based Detection Rate	89%	Detected 89% of malicious activity based on behavioral anomalies
Machine Learning Detection Rate	96–98%	Machine learning classifier performance across different flow types
Hybrid System Performance	Outperformed individual components	Combined voting method improved overall detection and adaptability
Resilience to Zero-Day Attacks	Slight performance drop	Maintained strong detection under adversarial and unknown conditions

6. DISCUSSION

This study shows that the hybrid approach to malware detection offers enhanced accuracy and efficiency compared to recent "best of breed" IoT security solutions. The proposed model's overall accuracy of 97.4% with a false positive rate of 2.3% suggests that the hybrid approach, which includes signature-based, anomaly-based and machine learning-based methods, considerably improves the effectiveness of malware detection in various scenarios.

The proposed model outperforms recent hybrid deep learning models. For example, the hybrid deep learning model proposed by Almazroi and Ayub [25] reported 96.1% accuracy, which is not as high as the results achieved in this research. Likewise, some of the previous rule-based hybrid systems such as Souza *et al*. [26] have shown 94% accuracy, but they also had low flexibility to adapt to new threats, indicating the need for a more adaptive and learning-based approach. The hybrid deep learning system proposed by Sahu *et al*. [27] reported an F1-score of 94.8%, which is also lower than the F1-score of the proposed hybrid model (96.2%).

The superiority of the proposed system is further substantiated by comparing with anomaly detection-based hybrid systems. While Ullah *et al*. [28] achieved an accuracy of 95.3%, they suffered from a higher false positive value (4.1%), whereas the proposed system improved detection by reducing false alarms and thus can be applied to IoT systems in real-time. Similarly, the AI-based model proposed by Alsubai *et al*. [29] reported 93.9% accuracy and was dependent on the size of the dataset and model interpretability, whereas the proposed model demonstrates better generalization with a balanced dataset and multi-layer detection approach.

For industrial IoT applications, Naeem *et al*. [30] proposed a hybrid deep learning image-based approach with 92.8% accuracy which is lower than the proposed model. Likewise, the hybrid deep learning model proposed by Rekha and Siddappa [31] showed high offline accuracy, without validating it in a real-time environment, as done in this work in a 48-hour IoT testbed. Another hybrid model based on the GhostNet architecture suggested by Almazroi and Ayub [32] could achieve 95.8% accuracy, which is still less than this study. Further, the hybrid SVM-rule-based system proposed by Ashraf *et al*. [33] reported 93.4% accuracy but is not capable of analysing behaviour, a key feature required to detect zero-day attacks.

Other recent work in deep learning-based systems also offers valuable insights. Ananthi *et al*. [34] exhibited good performance with deep learning models, but these models are heavy and may not be efficient for IoT devices. Also, hybrid machine learning models, such as Qureshi *et al*. [35], reinforce the benefits of using multiple detection techniques, as shown in this work. Efficient IDS proposed by Zainel [36] highlight the need for lightweight models, which are explored in this study through the use of Random Forest rather than complex deep learning models.

The growing complexity of threats, such as machine-generated malware as mentioned by Rustam *et al*. [37], also point to the need for versatile hybrid solutions. This work also uses a balanced dataset, which is consistent with benchmark datasets like ToN_IoT suggested by Alsaedi *et al*. [38], which highlight the need for realistic traffic representation. Furthermore, literature such as Doshi *et al*. [39] and Sarvari *et al*. [40] confirm the effectiveness of hybrid intrusion detection systems (anomaly and misuse) in improving detection performance, thus justifying the design of the proposed model.

Although deep learning-based IDS models [41], machine learning-based IDS [42] and other IDS methods have demonstrated effective detection, they are often inefficient and lack interpretability. However, efficient and edge-based IDS approaches [43] underline the need for lightweight models, which is realised in the proposed model. Similarly, extensive reviews such as Ferrag *et al*. [44] highlight that hybrid models are still the most promising to provide IoT security, due to their superior accuracy, adaptability, and scalability.

In summary, the hybrid framework in this study effectively overcomes major shortcomings of previous works such as high false positives, absence of real-time environment validation, and inefficiency. This research offers a scalable and effective approach to IoT security by integrating multiple detection methods and performing experiments in a real IoT environment.

7. CONCLUSIONS

With the rapid growth of Internet of Things (IoT) technologies in medical care, smart cities, and industrial automation, there is a greater risk of IoT systems being exposed to advanced cyber-attacks, including malware and botnets. Existing security strategies are often ineffective in these environments due to IoT device diversity and limitations. This requires the design of smart, adaptive and lightweight malware detection solutions. In this paper, we proposed a hybrid approach for malware detection combining signature-based detection, anomaly detection, and machine learning for classification. The empirical analysis has shown that the proposed model has a high detection accuracy of 97.4%, precision of 95.6%, recall of 96.8% and a false positive rate of 2.3%. Moreover, the system was tested in real-time over a 48-hour IoT testbed and showed promising results, with a detection rate of 97% and an average response time of 0.85 seconds. These results demonstrate the effectiveness of the proposed model in real-world IoT environments while achieving a high detection rate. In contrast to resource-inefficient deep learning models, the implementation of a Random Forest model achieved a balance between detection accuracy, interpretability and computational complexity, allowing the system to be deployed on low-powered IoT devices. Moreover, the use of multiple detection strategies improved the system's capability to detect both known and unknown attacks, showing good performance under attack scenarios. The research demonstrates the superiority of hybrid detection models over single detection models in terms of accuracy, resilience and adaptability to attacks. The combination of detection mechanisms overcomes limitations of current IoT security approaches, such as high false alarms and non-real-time responsiveness. To further improve the proposed model, future work should focus on incorporating explainable artificial intelligence (XAI) to improve model interpretability, adaptive response systems to mitigate detected threats in real-time and testing the framework on large-scale benchmark datasets, like ToN_IoT. Additionally, the use of edge computing for deployment and adaptive learning will enhance scalability and responsiveness in real-

world IoT settings. Overall, the proposed hybrid approach is a feasible and scalable method for improving IoT cybersecurity, and provides a base to develop next-generation smart intrusion detection systems to counter emerging cyber threats.

DECLARATION

Acknowledgement
The authors thank the College of Women's Sciences, Department of Computer Science, University of Babylon.

REFERENCES

[1] H. Wu, H. Han, X. Wang and S. Sun, "Research on Artificial Intelligence Enhancing Internet of Things Security: A Survey," in *IEEE Access*, vol. 8, pp. 153826-153848, 2020, https://doi.org/10.1109/ACCESS.2020.3018170.

[2] M. Abdullahi, Y. Baashar, H. Alhussian, A. Alwadain, N. Aziz, L. F. Capretz, and S. J. Abdulkadir, "Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review," *Electronics*, vol. 11, no. 2, p. 198, 2022, https://doi.org/10.3390/electronics11020198.

[3] G. Sadaram, M. Sakuru, L. M. Karaka, M. S. Reddy, V. Bodepudi, S. B. Boppana, and S. R. Maka, "Internet of things (IoT) cybersecurity enhancement through artificial intelligence: A study on intrusion detection systems," *Universal Library of Engineering Technology*, 2022, https://doi.org/10.70315/uloap.ulete.2022.001.

[4] I. Ahmad, Z. Wan, A. Ahmad, and S. S. Ullah, "A hybrid optimization model for efficient detection and classification of Malware in the internet of things," *Mathematics*, vol. 12, no. 10, p. 1437, 2024, https://doi.org/10.3390/math12101437.

[5] Z. E. Huma *et al*., "A Hybrid Deep Random Neural Network for Cyberattack Detection in the Industrial Internet of Things," in *IEEE Access*, vol. 9, pp. 55595-55605, 2021, https://doi.org/10.1109/ACCESS.2021.3071766.

[6] M. Y. Alzahrani and A. M. Bamhdi, "RETRACTED ARTICLE: Hybrid deep-learning model to detect botnet attacks over internet of things environments," *Soft Computing*, vol. 26, no. 16, pp. 7721-7735, 2022, https://doi.org/10.1007/s00500-022-06750-4.

[7] M. Nobakht, R. Javidan, and A. Pourebrahimi, "DEMD-IoT: a deep ensemble model for IoT malware detection using CNNs and network traffic," *Evolving Systems*, vol. 14, no. 3, pp. 461-477, 2023, https://doi.org/10.1007/s12530-022-09471-z.

[8] M. Ali, M. Shahroz, M. F. Mushtaq, S. Alfarhood, M. Safran and I. Ashraf, "Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment," in *IEEE Access*, vol. 12, pp. 40682-40699, 2024, https://doi.org/10.1109/ACCESS.2024.3376400.

[9] L. Almuqren, H. Alqahtani, S. S. Aljameel, A. S. Salama, I. Yaseen and A. A. Alneil, "Hybrid Metaheuristics With Machine Learning Based Botnet Detection in Cloud Assisted Internet of Things Environment," in *IEEE Access*, vol. 11, pp. 115668-115676, 2023, https://doi.org/10.1109/ACCESS.2023.3322369.

[10] S. Khan and A. B. Mailewa, "Discover botnets in IoT sensor networks: A lightweight deep learning framework with hybrid self-organizing maps," *Microprocessors and Microsystems*, vol. 97, p. 104753, 2023, https://doi.org/10.1016/j.micpro.2022.104753.

[11] S. Jain, P. M. Pawar, and R. Muthalagu, "Hybrid intelligent intrusion detection system for internet of things," *Telematics and Informatics Reports*, vol. 8, p. 100030, 2022, https://doi.org/10.1016/j.teler.2022.100030.

[12] J. Jeon, B. Jeong, S. Baek and Y. -S. Jeong, "Hybrid Malware Detection Based on Bi-LSTM and SPP-Net for Smart IoT," in *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4830-4837, 2022, https://doi.org/10.1109/TII.2021.3119778.

[13] R. Chaganti, V. Ravi, and T. D. Pham, "Deep learning based cross architecture internet of things malware detection and classification," *Computers & Security*, vol. 120, p. 102779, 2022, https://doi.org/10.1016/j.cose.2022.102779.

[14] H. A. Alterazi *et al*., "Prevention of cyber security with the internet of things using particle swarm optimization," *Sensors*, vol. 22, no. 16, p. 6117, 2022, https://doi.org/10.3390/s22166117.

[15] N. Ahmed, M. A. Ngadi, A. A. Almazroi, and N. A. Alghanmi, "Hybrid model for novel attack detection using a cluster-based machine learning classification approach for the Internet of Things (IoT)," *Future Internet*, vol. 17, no. 6, p. 251, 2025, https://doi.org/10.3390/fi17060251.

[16] S. Kasarapu, S. Shukla, and S. M. P. Dinakarrao, "Enhancing IoT malware detection through adaptive model parallelism and resource optimization," *arXiv preprint arXiv:2404.08808*, 2024, https://doi.org/10.48550/arXiv.2404.08808.

[17] A. Mehrban and P. Ahadian, "Malware detection in iot systems using machine learning techniques," *arXiv preprint arXiv:2312.17683*, 2023, https://doi.org/10.48550/arXiv.2312.17683.

[18] S. H. Khan *et al*., "A new deep boosted CNN and ensemble learning based IoT malware detection," *Computers & Security*, vol. 133, p. 103385, 2023, https://doi.org/10.1016/j.cose.2023.103385.

[19] M. Asam *et al*., "IoT malware detection architecture using a novel channel boosted and squeezed CNN," *Scientific Reports*, vol. 12, no. 1, p. 15498, 2022, https://doi.org/10.1038/s41598-022-18936-9.

[20] S. K. Smmarwar, G. P. Gupta, and S. Kumar, "AI-empowered malware detection system for industrial internet of things," *Computers and Electrical Engineering*, vol. 108, p. 108731, 2023, https://doi.org/10.1016/j.compeleceng.2023.108731.

[21] M. Abdullahi *et al*., "Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review," *Electronics*, vol. 11, no. 2, p. 198, 2022, https://doi.org/10.3390/electronics11020198.
[22] J. Jeon, B. Jeong, S. Baek and Y. -S. Jeong, "Static Multi Feature-Based Malware Detection Using Multi SPP-net in Smart IoT Environments," in *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 2487-2500, 2024, https://doi.org/10.1109/TIFS.2024.3350379.
[23] M. J. Kathem and T. S. Atia, "A Review on IoT Cyber-Attacks Detection Challenges and Solutions," *Al-Iraqia Journal for Scientific Engineering Research*, vol. 2, no. 3, pp. 22-31, 2023, https://doi.org/10.58564/IJSER.2.3.2023.84.
[24] R. A. Zidan and G. Karraz, "Towards An Efficient Internet of Things Intrusion Detection by Using Support Vector Machine," *Baghdad Science Journal*, vol. 22, no. 5, pp. 1714-1724, 2025, https://doi.org/10.21123/bsj.2024.11067.
[25] A. A. Almazroi and N. Ayub, "Deep learning hybridization for improved malware detection in smart Internet of Things," *Scientific reports*, vol. 14, no. 1, p. 7838, 2024, https://doi.org/10.1038/s41598-024-57864-8.
[26] P. V. de C. Souza, A. J. Guimarães, T. S. Rezende, V. Souza Araujo, L. A. F. do Nascimento and L. Oliveira Batista, "An Intelligent Hybrid Model for the Construction of Expert Systems in Malware Detection," *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pp. 1-8, 2020, https://doi.org/10.1109/EAIS48028.2020.9122770.
[27] A. K. Sahu, S. Sharma, M. Tanveer, and R. Raja, "Internet of Things attack detection using hybrid Deep Learning Model," *Computer Communications*, vol. 176, pp. 146-154, 2021, https://doi.org/10.1016/j.comcom.2021.05.024.
[28] I. Ullah, A. Ullah, and M. Sajjad, "Towards a hybrid deep learning model for anomalous activities detection in internet of things networks," *IoT*, vol. 2, no. 3, pp. 428-448, 2021, https://doi.org/10.3390/iot2030022.
[29] S. Alsubai, A. K. Dutta, A. M. Alnajim, R. Ayub, A. M. AlShehri, and N. Ahmad, "Artificial intelligence-driven malware detection framework for internet of things environment," *PeerJ Computer Science*, vol. 9, p. e1366, 2023, https://doi.org/10.7717/peerj-cs.1366.
[30] H. Naeem *et al*., "Malware detection in industrial internet of things based on hybrid image visualization and deep learning model," *Ad Hoc Networks*, vol. 105, p. 102154, 2020, https://doi.org/10.1016/j.adhoc.2020.102154.
[31] H. Rekha and M. Siddappa, "Hybrid deep learning model for attack detection in internet of things," *Service Oriented Computing and Applications*, vol. 16, no. 4, pp. 293-312, 2022, https://doi.org/10.1007/s11761-022-00342-8.
[32] A. A. Almazroi and N. Ayub, "Enhancing smart IoT malware detection: A GhostNet-based hybrid approach," *Systems*, vol. 11, no. 11, p. 547, 2023, https://doi.org/10.3390/systems11110547.
[33] M. W. A. Ashraf, A. R. Singh, A. Pandian, R. S. Rathore, M. Bajaj, and I. Zaitsev, "A hybrid approach using support vector machine rule-based system: detecting cyber threats in internet of things," *Scientific Reports*, vol. 14, no. 1, p. 27058, 2024, https://doi.org/10.1038/s41598-024-78976-1.
[34] P. Ananthi, K. Nirmaladevi and M. G, "Deep Learning Based Intrusion Detection for IoT Networks," *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*, pp. 1690-1694, 2024, https://doi.org/10.1109/ICUIS64676.2024.10866629.
[35] S. U. Qureshi *et al*., "Systematic review of deep learning solutions for malware detection and forensic analysis in IoT," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 8, p. 102164, 2024, https://doi.org/10.1016/j.jksuci.2024.102164.
[36] H. A. Zainel, "Lightweight Federated Intrusion Detection System for Resource Constrained IoT Networks Using Edge-Assisted Learning," *Journal of Al-Turath University College*, vol. 43, no. 1, pp. 388-405, 2026, https://doi.org/10.63964/JATUC.43.1.2026.32.
[37] F. Rustam, P. Ranaweera and A. D. Jurcut, "AI on the Defensive and Offensive: Securing Multi-Environment Networks from AI Agents," *ICC 2024 - IEEE International Conference on Communications*, pp. 4287-4292, 2024, https://doi.org/10.1109/ICC51166.2024.10622943.
[38] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," in *IEEE Access*, vol. 8, pp. 165130-165150, 2020, https://doi.org/10.1109/ACCESS.2020.3022862.
[39] R. Doshi, N. Apthorpe and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 29-35, 2018, https://doi.org/10.1109/SPW.2018.00013.
[40] S. Sarvari, N. F. Mohd Sani, Z. Mohd Hanapi and M. T. Abdullah, "An Efficient Anomaly Intrusion Detection Method With Feature Selection and Evolutionary Neural Network," in *IEEE Access*, vol. 8, pp. 70651-70663, 2020, https://doi.org/10.1109/ACCESS.2020.2986217.
[41] C. Yin, Y. Zhu, J. Fei and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," in *IEEE Access*, vol. 5, pp. 21954-21961, 2017, https://doi.org/10.1109/ACCESS.2017.2762418.
[42] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for IoT applications," *arXiv preprint arXiv:2302.12452*, 2023, https://doi.org/10.48550/arXiv.2302.12452.
[43] H. G. A. Umar *et al*., "Energy-efficient deep learning-based intrusion detection system for edge computing: a novel DNN-KDQ model," *Journal of Cloud Computing*, vol. 14, no. 1, p. 32, 2025, https://doi.org/10.1186/s13677-025-00762-9.
[44] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020, https://doi.org/10.1016/j.jisa.2019.102419.

AUTHOR BIOGRAPHY

Aseel Hamoud Hamza, was born in Hillah, Babylon. She received her bachelor's degree from the University of Babylon, College of Information Technology, and completed her master's studies from the University of Babylon, College of Women's Sciences, Department of Computer Science, in the field of information security, from 2017 to 2019, and she worked in the field of teaching from 2019 to 2022 in University of Babylon.

Rusul H. Altaie, was born in Babylon city, Iraq in 1988. She received the B.S. degree in computer science from the University of Babylon, in 2010, M.S. degree in information technology from the University of Babylon in 2016 and the Ph.D. degree in information technology from University of Babylon, in 2024. From 2012 to 2016, she was a Programmer Assistant in the computer Laboratory. Her research interests include Internet of Thing, security, Artificial Intelligence, Image Processing and Intrusion Detection.