Quantization-Aware Training for Man-in-the-Middle Attacks Detection in IoT Application

Dyah Ayu Suci Ilhami[1], Aji Gautama Putrada[2], and Farah Afianti[3]

***Abstract*—Issues in securing the resilience of Deep Learning models and operational problems related to model size and latency in the Internet of Things (IoT) environment, especially when applied to devices with limited resources, can be overcome by implementing a modified Quantization-Aware Training (QAT) model based on IDS and an autoencoder. This research proposes the use of QAT to achieve operational efficiency while strengthening the detection model's resilience against data manipulation in MiTM attacks. By integrating QAT, the model not only becomes lighter for edge devices but also more capable of maintaining detection integrity even when model weights are compromised. The research was conducted using the CICIoT2023 dataset, and the output of the advanced QAT model was then applied to resource-constrained IoT devices, namely the Raspberry Pi 3. The methodology began with data collection, followed by data preprocessing, which was then normalized before being fed into the IDS and autoencoder techniques. After the autoencoder model was successfully created, the QAT model was developed so that it remained unchanged or even improved when implemented on the Raspberry Pi 3. With the application of QAT modifications, an 80.43% reduction in model size and an 11.2% increase in model inference speed were confirmed. Furthermore, when faced with QAT model weight corruption of up to 20%, the F1-score value remained stable at 100%. These results show that the QAT model is highly effective at improving model reliability and maintaining the resilience of deep learning models against MiTM attacks, even under limited conditions.**

I. INTRODUCTION

In the Internet of Things (IoT) environment, with massive volumes of data and various techniques used to prevent attacks—such as machine learning (ML), threshold-based techniques, edge computing, and deep learning (DL) [1]. DL and ML techniques can also be applied to data management in IoT to mitigate attacks such as DOS, DDoS, ransomware, man-in-the-middle (MiTM), and zero-day attacks [2]. This research will focus on MiTM attacks, which can manipulate input data and reduce prediction accuracy, thereby compromising model integrity [2]. Generally, a MiTM attacker positions themselves between an IoT sensor and a gateway to modify data traffic, with the primary challenge being how the system can detect such attacks in real time before the attacker successfully manipulates the system. There is little research focused on MiTM attacks, making this a significant issue in resource-constrained IoT environments that must operate with high latency and integrity [3]. Therefore, this research assumes that detection within IoT can minimize response latency.

Several techniques have been designed, such as Quantized Auto Encoder Detector (QAED), Deep Neural Networks (DNN), Quantization Aware Training (QAT), and Quantized Autoencoder (QAE) [4]. Each technique has its own advantages and disadvantages, as in the research by Trappolini, which aims to maintain DL resilience while detecting intrusions using a QAED model [5]. This successfully improves accuracy and efficiency in anomaly detection using the SOTA method but requires significant computing power and intensive training time. The parameters tested are also too broad when applied to limited computational resources. [5]

In another study by Rahmati, the resilience of DL on IoT devices was tested using the QAT technique. The results showed that the QAT model significantly reduced energy consumption by 35.4% and improved resilience against attacks on the dataset [6]. In addition, Hernandez's research used two quantization techniques, namely PTQ and QAT, and showed that QAT, when applied to IoT, achieved much higher energy efficiency and demonstrated its feasibility for overcoming CNN-based face recognition attacks in IoT devices [7]. Compared to the research by Trappolini using QAED and Rahmati or Hernandez using QAT for their quantization techniques, QAT is considered better at improving efficiency without high complexity when implemented on resource-limited computing devices.

The selection of QAT techniques for detecting MiTM attacks is crucial due to security requirements in the IoT ecosystem. According to research by Jena et al. [8], quantization techniques are highly effective in reducing latency without drastically compromising accuracy. Furthermore, research conducted by Michelena [9] demonstrates that QAT can enhance the resilience of detection models against attacks by strengthening the integrity of model weights during the training process. Therefore, QAT serves not only as a tool for efficiency optimization but also as a defense mechanism that ensures detection remains fast and accurate under the pressure of integrity attacks.

Therefore, this research will use a modified QAT model tailored to the research objectives. This study will utilize the CICIoT2023 dataset, which contains both normal and anomalous data. However, this research will focus primarily on the analysis of imbalanced datasets; thus, in the initial stage, the imbalanced dataset will be processed using an Intrusion Detection System (IDS). Next, the dataset will be trained using

*This work was not supported by any organization

[1]Dyah Ayu Suci Ilhami Author in School of Computing, Telkom University, Bandung, Indonesia dyahayusuciilhami@student.telkomuniversity.ac.id

[2]Aji Gautama Putrada Author in School of Computing, Telkom University, Bandung, Indonesia ajigps@telkomuniversity.ac.id

[3]Farah Afianti Author in School of Computing, Telkom University, Bandung, Indonesia farahafi@telkomuniversity.ac.id

Manuscript received January 9, 2026; accepted April 8, 2026.

the autoencoder technique to obtain a baseline AUC value from the ROC curve of ≥ 0.8. If this condition is met, the QAT technique will proceed. The QAT technique is used to reduce the model size, thereby enabling optimal evaluation and analysis on the Raspberry Pi

To the best of our knowledge, no research studies have examined the use of QAT to detect MITM attacks on IoT systems. The contributions of this research are as follows:

1) Validation of the QAT model, which integrates security performance stability and operational efficiency on a Raspberry Pi 3 device using the comprehensive CICIoT2023 dataset.
2) A knowledge that QAT can improve model integrity, even when model weights are corrupted by up to 20%, to mitigate MITM attack threats.
3) An anomaly detection on IoT devices with the CICIoT2023 dataset using a QAT model configuured by IDS.

The contents of the research are designed as follows: Section II reviews related work, Section III describes the methodology, Section IV discusses the experimental results, and Section VI concludes the paper.

II. Related Works

Various research has explored anomaly detection methods to improve the security of IoT ecosystems. Al-Ghamdi applied CNN and RNN techniques to the CICIoT2023 dataset and achieved an initial accuracy of 98%, although the final results varied depending on the specific techniques used [10]. Meanwhile, Khan focused on anomaly detection in the healthcare sector using RF and DNN on the same dataset; however, the study did not specifically address MiTM attacks [11]. Next, Kandasamy developed a real-time detection system using AE and EXB fusion techniques, which provides rapid response but requires significant computational resources [12]. Additionally, Michelena investigates MiTM detection using the MQTT protocol; however, the method remains limited by high memory and resource consumption [9].

On the other side, there have been several researches on operational efficiency in resource-constrained devices that already employ quantization techniques. Research conducted by Trappolini [5] introduced the Quantized Auto Encoder Detector (QAED) to improve intrusion detection efficiency, although this technique faces challenges such as long training times and high computational requirements. Furthermore, research by Jena [8] evaluates PTQ and QAT techniques using the MVTec AD dataset, showing that QAT can maintain accuracy while reducing latency, although its training process is longer. Although various quantization techniques have been proposed for efficiency, the integration of QAT with comprehensive datasets such as CICIoT2023 to protect model integrity against weight manipulation caused by MiTM remains an under-explored research opportunity. Table I summarizes how this research addresses the identified gaps compared to previous research. Meanwhile, Gheni and Al-Yaseen [13] validated an intrusion detection system (IDS) using the same CICIoT2023 dataset and achieved high efficiency by reducing the data dimension by 62.45% without compromising detection accuracy against various attacks, including spoofing

TABLE I: State-of-the-Art Research on IoT Anomaly Detection, Quantization-Aware Training, and MitM Attack Detection

Paper	CICIoT2023 Dataset	Quantization	MiTM Attack
[10]	✓	✗	✗
[11]	✓	✗	✗
[5]	✗	✓	✗
[8]	✗	✓	✗
[12]	✗	✗	✓
[9]	✗	✗	✓
[13]	✓	✓	✗
Proposed Thesis	✓	✓	✓

III. Proposed Method

The research workflow is designed. It starts with the CICIoT2023 dataset obtained from the Canadian Institute for Cybersecurity, followed by feature normalization and encoding. An autoencoder model is built and trained for anomaly detection, which is measured using ROC and AUC. Then, QAT is applied to the model to enable its implementation on Raspberry Pi 3 and to improve its resistance to integrity attacks. Finally, the results were evaluated based on detection performance, resource efficiency (model size and latency), and improved integrity compared to the full-precision baseline. Fig. 1 shows the research workflow.

A. Dataset Selection and Characteristics

For the application of the proposed QAT method in this research, a comprehensive dataset from CIC will be used, namely the CICIoT2023 dataset, publicly available repository designed for IoT cybersecurity research. This dataset is the latest and most extensive collection for IoT cybersecurity research [10], collected from the IoT laboratory network at the Canadian Institute for Cybersecurity, and covers network

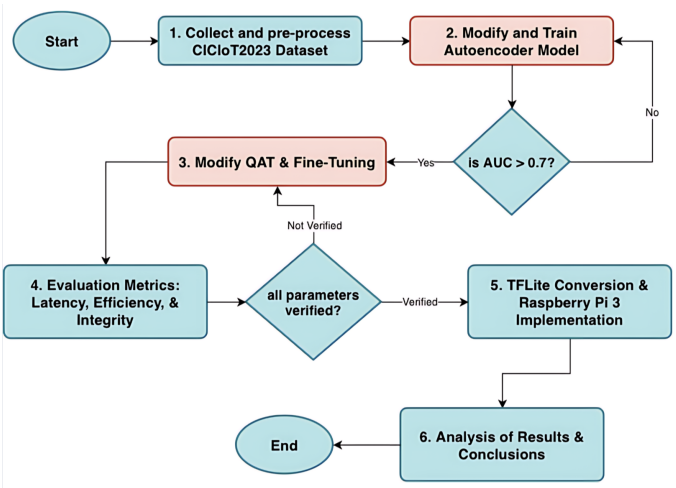

Fig. 1: Workflow Diagram for Enhancing Model Integrity Against Man-in-the-Middle Attacks

activity from 105 IoT devices across 35 attack types in seven main categories. Although it involves many attacks, this research focuses on modifying the model to withstand MiTM attacks that threaten integrity through data manipulation or unauthorized models.

The main attack categories displayed, aside from MITM, are DDoS, DoS, Recon, Web-Based Attacks, Brute Force Attacks, Spoofing, and Mirai. Critically, attacks such as Spoofing directly represent integrity threats (MITM) that QAT targets by securing models at the network edge. This dataset contains tens of thousands of data rows, available in PCAP and CSV formats [14], and has been processed for model training.

B. Modifying Autoencoder Model

After the raw dataset is successfully collected, it will be processed through a data preprocessing stage before being used and applied to the model. The preprocessing stage encodes all non-numeric features in the dataset [15].This preprocessing process converts raw CSV data into numerical features by applying label encoding to categorical data. Of the 46 available features, specific flow metrics such as 'flow duration' and 'packet length' are normalized via Min-Max Scaling to the range [0,1]. This ensures that features with extremely large values do not dominate loss calculations during autoencoder training. After all data has been successfully normalized, the dataset is divided into three sets: training (70%), validation (15%), and testing (15%) [16]. The normalized training data is then used with the autoencoder to reconfirm anomaly detection on 70% of the data. By applying the autoencoder technique, the model can effectively identify threats and monitor that when anomalous input is detected, it will immediately produce a high reconstruction value [17].

The autoencoder (AE) implemented in this research uses a symmetric architecture consisting of an encoder with three layers and a decoder with three layers [18]. Each layer maps vectors; for example, the encoder layer maps 46 dimensions to a bottleneck (latent) space of 8 neurons. The structure in the autoencoder evolves from 46 dimensions $\rightarrow$ 32 dimensions $\rightarrow$ 16 dimensions $\rightarrow$ 8 dimensions, which is then reflected to 46 dimensions, as described in detail in Table II [19]. After the model was successfully built, to minimize the difference between the input and the reconstructed output, we used the Mean Squared Error (MSE) loss function and the Adam optimizer [20].

TABLE II: Detailed AE Architecture

Layer	Output	Act.	Param
input_layer	(None, 46)	N/A	0
Encoder			
dense	(None, 32)	ReLU	1,504
dense_1	(None, 16)	ReLU	528
dense_2 (Bottleneck)	(None, 8)	ReLU	136
Decoder			
dense_3	(None, 16)	ReLU	144
dense_4	(None, 32)	ReLU	544
dense_5 (Output)	(None, 46)	Lin/Sig*	1,518
		Total Params:	**4,374**

C. Modifying QAT Model and Raspberry Pi Implementation

If the autoencoder architecture has been successfully created, proceed to the QAT model modification stage. The QAT model is used to optimize the model by compressing its size so that the functions and model size requirements built previously can be applied to IoT devices, such as the Raspberry Pi 3 [8]. The application of QAT can also obtain optimal values without reducing model accuracy [21].

The QAT model works by inserting "fake quantization" nodes during training to simulate the rounding effects of converting from 32-bit floating-point (FP32) precision to 8-bit integer (INT8) precision [22]. This process allows the model to adjust its weights to remain accurate even with reduced precision. The mapping equation used to convert continuous values to discrete values is:

$$q = \text{round}\left(\frac{r}{S} + Z\right) \tag{1}$$

where r represents a real-valued input, S represents the scale factor, and Z is a zero-point integer. By integrating this simulation into the forward pass, the model can minimize the quantization error that typically occurs in standard post-training methods [8]. The process of applying the QAT model algorithm, from input and output details to the implementation, is shown in Algorithm 1.

Algorithm 1 Quantization-Aware Training and Deployment

Require: Pre-trained Model M_{FP32} with weights w, Training Data D_{train}, Optimization Settings (Adam, L_{MSE}), Fine-tuning Epochs E_q, Target Hardware
Ensure: Lightweight INT8 TFLite Model M_{INT8}
1: **Initialize:** Load the pre-trained baseline model M_{FP32}
2: **Insert Fake Quantization Nodes:** $Q(\cdot)$ for weight tensors $w_q = Q(w)$ and activation tensors $a_q = Q(a)$ using $q = \text{round}(r/S + Z)$
3: **Quantization-Aware Training:**
4: **for** $e = 1$ to E_q **do**
5: Feed a minibatch from D_{train} into the modified model
6: Compute reconstruction loss $L_{MSE}(x, \hat{x})$
7: Update model weights w using Adam optimizer: $w_{t+1} = w_t - \eta \cdot \nabla L_{MSE}$, where weights are constrained to simulated INT8 ranges during backpropagation
8: **end for**
9: Save the fine-tuned model M_{QAT_FP32}
10: **Final Quantization and Conversion:**
11: $M_{QAT\_FP32} \leftarrow$ Remove $Q(\cdot)$ and freeze statistics (S, Z)
12: $M_{INT8} \leftarrow$ Map w_q and a_q to 8-bit integer range $[-128, 127]$
13: **Deployment:** Deploy M_{INT8} on the target hardware
14: **return** M_{INT8}

After the deployment process defined in 1 is complete, the next step is to evaluate the required metrics. The evaluation of metrics is based on three parameters, namely Detection Performance, IoT Efficiency, and Model Integrity. Looking at the first parameter, detection performance, the evaluation metric is the AUC obtained with the autoencoder and after applying QAT [23]. Next, the second parameter for IoT

efficiency is evaluated by comparing the autoencoder's model size to that of the QAT model implementation. The third parameter, Model Integrity, is assessed by conducting various experiments in which we randomly corrupt and insert into the model's weights, then evaluate the model's performance on the autoencoder or baseline, and finally implement the QAT model [24].

After evaluating the metrics, we moved on to the final stage of the research: applying the model to a device with limited resources. In this study, we used a Raspberry Pi 3 IoT device. The Raspberry Pi 3 device used is the Model B with 1 GB of RAM [25], running the Raspberry Pi OS. The evaluation was conducted in an offline environment [26], where the pre-processed TFLite model was loaded locally on the device to measure inference latency and model size without any network-induced delays. Before applying the model to the Raspberry Pi 3, we had to convert it to an acceptable format, namely TensorFlow Lite (TFLite), due to its limitations [27]. The TFLite format was chosen because of its very small size, which results in shorter processing times and lower resource consumption. This makes it suitable for implementation on the Raspberry Pi 3, ensuring that the hardware is not overloaded when running the developed model [25].

IV. Results and Discussion

A. Result

The research process involved collecting the CICIoT2023 dataset directly from the public dataset released by the Canadian Institute for Cybersecurity (CIC). The dataset includes a series of network activities from 105 IoT devices, including 35 attack types categorized by CIC into 7 main attack categories. The dataset then underwent data preprocessing, during which all features were digitized using the encoding function and successfully normalized using min-max scaling. The normalized dataset was then divided into 70% for training (307,715 rows), 15% for validation (65,939 rows), and 15% for testing (65,939 rows).

Next, build an autoencoder model using 70% of the training data. Once the model has been successfully created, check its performance to see if there is any anomaly detection from intrusions using the ROC curve graph in Fig. 2. The ROC curve shows that the autoencoder model is at the optimal threshold, with an Equal Error Rate (EER) of 0.2290 and a fairly high F1-score of 0.7410. The high performance of the autoencoder model validates its suitability as a foundation for the next step in the Quantization Aware Training (QAT) stage.

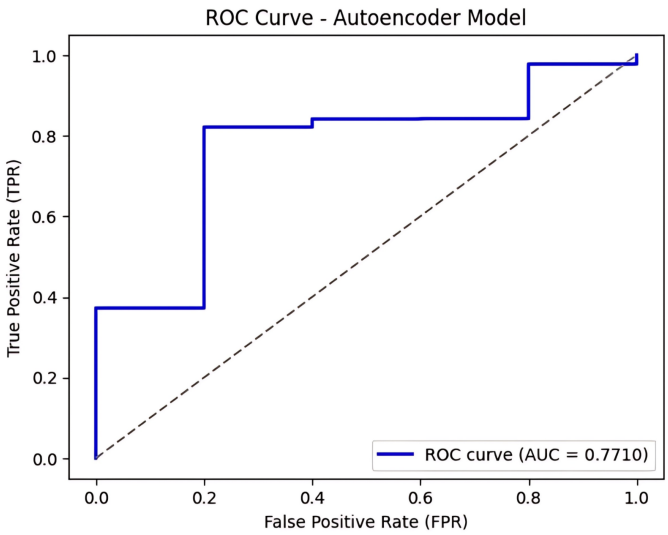

Fig. 2: Baseline ROC curve.

After the foundation was successfully created from the autoencoder model, the quantization-aware technique (QAT) was applied. Then, a fake quantization node was inserted into the autoencoder model while continuing to optimize encoding and scaling. Then, we re-measure the Area Under the Curve value after QAT is successfully applied (see Fig. 3). The AUC value increases from 0.7710 before QAT to 0.8100 after QAT, showing an increase of 5.06%.

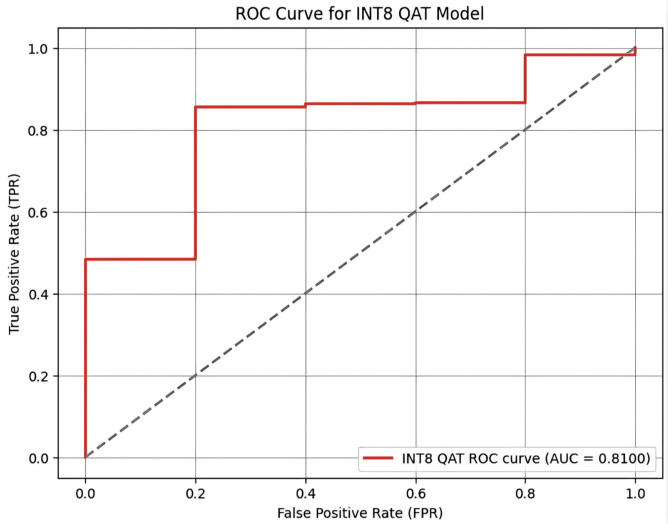

Fig. 3: ROC Curve After QAT Model Implemented

As shown in Table III, after QAT was successfully applied, the model's performance improved and remained stable. As with the AUC value, which increased, the EER value improved by 12.66%, and the F1-Score value decreased minimally by 0.01%. This QAT value validates that the QAT model can function as a performance enhancer and efficiency stabilizer under limited conditions. This summary of values confirms that the QAT model, which was successfully built, functions as an efficiency stabilizer and performance enhancer.

TABLE III: Comparative Performance (Baseline vs QAT Model)

Metric	Base.	QAT	Diff.	Change (%)
AUC	0.7710	**0.8100**	**+0.0390**	**+5.06%**
F1-Score	**0.7410**	0.7409	-0.0001	-0.01%
EER (Equil.)	0.2290	**0.2000**	-0.0290	-12.66%
FAR/FRR	0.2290	**0.2000**	-0.0290	-12.66%

In addition to the AUC value and several metrics that improved and stabilized after applying the QAT model, the model size was successfully reduced from 95.15 KB to 12.64 KB, representing an 80.43% decrease. More complete data are shown in Table IV. The reduction in file size and inference latency, as shown in the table, enables the model to be implemented on target devices with limited resources, namely the Raspberry Pi 3. With an efficiency improvement of 80.43% after deployment on the Raspberry Pi 3 without sacrificing accuracy, this validates the feasibility of the QAT technique for similar security deployments on limited-resource computing devices.

TABLE IV: IoT Efficiency Comparison on Raspberry Pi 3

Metric	FP32	INT8 (QAT)	Red./Imp.
Model Size (KB)	95.16	**12.64**	80.43%
Latency (ms)	0.0965	**0.0042**	95.66%

B. Discussion

Based on the results of the experiments that have been conducted, the first contribution, the dual benefits of the QAT technique, has successfully achieved maximum results. This is evidenced by an 80.43% reduction in model size after successful QAT implementation, as well as an 11.2% increase in latency when the model was deployed on a Raspberry Pi 3. The efficiency of this significant research is very different from the results of research conducted by Trappolini, who detected IDs using the QAED technique [5]. Trappolini improved the accuracy of his study but required high computing power, making it impossible to test its accuracy when implemented on limited-resource devices [5]. In contrast, this study achieved efficiency without sacrificing prior performance. Thus, this reinforces the study's first contribution, which is to validate the dual benefits of QAT: performance stability and operational efficiency.

Continuing with the experimental results for the second contribution to the research, which is a demonstration that successfully shows that QAT can improve model integrity by corrupting model weights. This is demonstrated by the F1-score values when the QAT technique is applied across 0% to 20% weight destruction, all of which remain at 100%, indicating that the QAT model can still maintain its integrity even when its weights are destroyed. When compared to the research conducted by Kandasamy, which achieved high accuracy and real-time detection capabilities with fast responses in detecting MiTM attacks using AE and EXB [12]. However, Kandamsy's research is highly dependent on the dataset and the parameters within it, so the results cannot be proven; if the model's parameters are changed, the accuracy and response time will remain fast [12]. Thus, the QAT model here can reinforce the second contribution of the research, namely, maintaining model integrity even when the space is forced to be compromised by MiTM attacks.

The third contribution to the research is the successful implementation of QAT-based IDS. The results of the experiment show that the QAT technique is capable of overcoming the challenges of comprehensive security evaluation of this IDS. This is because many previous studies have ignored IDS when using quantization techniques. As in the research conducted by Wang, who used the Dynamic Quantization technique, which developed a method to improve efficiency for computational resource devices (IoT) by focusing on DoS, DDoS, and Mirai attacks [21]. However, his IDS system could only identify patterns and behaviors that were not too complex, due to the limitations of the techniques he used [21]. In contrast, this study, by applying QAT to a comprehensive dataset such as the CICIoT2023 dataset, highlighted the third contribution: the IDS applied to the QAT model successfully validated the model's robustness. Compared to other optimization strategies such as pruning or knowledge distillation, which prioritize model size reduction [21], the QAT approach specifically addresses the trade-off between safety and efficiency. Different from standard pruning, which removes important neurons related to security, QAT fine-tuning maintains an F1 score of 100% even with a 20% weight loss. This demonstrates that QAT is more effective in detecting MiTM attacks because it preserves model integrity while achieving the necessary computational efficiency

V. Limitations

While this research successfully validates QAT as a solution for both efficiency and integrity, several constraints remain. First, the evaluation was conducted on a Raspberry Pi 3, which is a single-CPU-based device; thus, the results may vary on edge devices equipped with dedicated hardware accelerators like NPUs or TPUs [26]. Second, the model integrity testing was limited to simulated weight corruption representing MiTM attacks. Future studies should expand this by testing against hardware-level attacks and activation-layer manipulation to provide a more comprehensive security profile [24]. Lastly, this study focused on offline model strengthening. Integrating a real-time integrity verification layer would be a valuable direction for future research to enable active detection and response in live IoT deployments.

VI. Conclusion

This research successfully validated the Quantization-Aware Training (QAT) model as a crucial technique for maintaining Deep Learning (DL) resilience. This technique is vital in improving the security, integrity, and operational efficiency of autoencoder models. The QAT model in this study successfully provided an optimal solution, reducing model size by 80.43% (from 95.16 KB to 12.64 KB). These QAT results provide a solution for implementing DL model resilience against attacks on limited-power IoT devices such as the Raspberry Pi 3. In addition, the QAT model increased efficiency and performance

from 0.7710 to 0.8100. Even when the QAT model was deliberately corrupted by 20%, the F1 score remained stable with resistance to man-in-the-middle (MiTM) attacks. Overall, the QAT model effectively ensures the implementation of security models and effectively limits the space for manipulation in resource-constrained IoT environments.

For future research, there are several possibilities, such as applying QAT on more advanced hardware, testing model integrity with additional activation manipulation techniques, and verifying model integrity in real time on other IoT devices.

ACKNOWLEDGMENT

The authors would like to thank the Canadian Institute for Cyber Security (CIC) for creating and publishing the CICIoT2023 dataset, which was very helpful in the proposed research. Furthermore, we would like to thank the Directorate of Research and Community Service (PPM) at Telkom University for its support in infrastructure and funding, which facilitated the completion of this research.

REFERENCES

[1] M. Zaman, N. Puryear, S. Abdelwahed, and N. Zohrabi, "A review of iot-based smart city development and management," *Smart Cities*, vol. 7, no. 3, pp. 1462–1501, 2024.

[2] M. M. Cherian and S. L. Varma, "Mitigation of ddos and mitm attacks using belief based secure correlation approach in sdn-based iot networks," *International Journal of Computer Network and Information Security*, vol. 15, no. 1, p. 52, 2022.

[3] H. Fereidouni, O. Fadeitcheva, and M. Zalai, "Iot and man-in-the-middle attacks," *Security and Privacy*, vol. 8, no. 2, p. e70016, 2025.

[4] S. Ryu, J. Yim, J. Seo, Y. Yu, and H. Seo, "Quantile autoencoder with abnormality accumulation for anomaly detection of multivariate sensor data," *IEEE Access*, vol. 10, pp. 70428–70439, 2022.

[5] G. Trappolini, A. Purificato, F. Siciliano, L. D'Addona, A. M. Spagnolo, D. Dato, and F. Silvestri, "Quantized auto encoder-based anomaly detection for multivariate time series data in 5g networks," *IEEE Access*, 2025.

[6] M. Rahmati, "Energy-aware federated learning for secure edge computing in 5g-enabled iot networks," *Journal of Electrical Systems and Information Technology*, vol. 12, no. 1, p. 13, 2025.

[7] E. Hernandez-Laredo, Á. G. Estévez-Pedraza, L. M. Santiago-Fuentes, and L. Parra-Rodríguez, "Optimizing fall risk diagnosis in older adults using a bayesian classifier and simulated annealing," *Bioengineering*, vol. 11, no. 9, p. 908, 2024.

[8] S. Jena, A. Pulkit, K. Singh, A. Banerjee, S. Joshi, A. Ganesh, D. Singh, and A. Bhavsar, "Unified anomaly detection methods on edge device using knowledge distillation and quantization," in *International Workshop on Reproducible Research in Pattern Recognition*, pp. 60–81, Springer, 2024.

[9] A. Michelena, J. Aveleira-Mata, E. Jove, M. Bayón-Gutiérrez, P. Novais, O. F. Romero, J. L. Calvo-Rolle, and H. Aláiz-Moretón, "A novel intelligent approach for man-in-the-middle attacks detection over internet of things environments based on message queuing telemetry transport," *Expert Systems*, vol. 41, no. 2, p. e13263, 2024.

[10] A. M. Al-Ghamdi and M. M. Alansari, "Enhancing iot security: A comparative study of cnn and rnn-based anomaly detection using the ciciot2023 dataset.," *IAENG International Journal of Computer Science*, vol. 52, no. 5, 2025.

[11] M. M. Khan and M. Alkhathami, "Anomaly detection in iot-based healthcare: machine learning for enhanced security," *Scientific Reports*, vol. 14, no. 1, p. 5872, 2024.

[12] V. Kandasamy and A. A. Roseline, "Harnessing advanced hybrid deep learning model for real-time detection and prevention of man-in-the-middle cyber attacks," *Scientific Reports*, vol. 15, no. 1, p. 1697, 2025.

[13] H. Q. Gheni and W. L. Al-Yaseen, "Two-step data clustering for improved intrusion detection system using cic-iot2023 dataset," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 9, p. 100673, 2024.

[14] M. Sliwka, M. Wojcik, and B. Kurek, "A comparative analysis of anomaly detection methods in iot networks: An experimental study," *Sensors*, vol. 14, no. 24, p. 11545, 2024.

[15] Y. Li, Y. Qian, Y. Yu, X. Qin, C. Zhang, Y. Liu, K. Yao, J. Han, J. Liu, and E. Ding, "Structext: Structured text understanding with multi-modal transformers," in *Proceedings of the 29th ACM international conference on multimedia*, pp. 1912–1920, 2021.

[16] I. Muraina, "Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts," in *7th international Mardin Artuklu scientific research conference*, pp. 496–504, 2022.

[17] A. Hassan, H. Khan, I. Uddin, A. Sajid, *et al.*, "Optimal emerging trends of deep learning technique for detection based on convolutional neural network," *Bulletin of Business and Economics (BBE)*, vol. 12, no. 4, pp. 264–273, 2023.

[18] K. Yu, J. Zhu, W. Yuan, Q. Zhou, G. Xue, G. Wu, X. Wang, and X. Li, "Two-channel six degrees of freedom grating-encoder for precision-positioning of sub-components in synthetic-aperture optics," *Optics express*, vol. 29, no. 14, pp. 21113–21128, 2021.

[19] A. Ashraf, N. M. Nawi, T. Shahzad, M. Aamir, M. A. Khan, and K. Ouahada, "Dimension reduction using dual-featured auto-encoder for the histological classification of human lungs tissues," *IEEE access*, 2024.

[20] X. Zhang, J. Wu, F. Zhang, Z. Lin, and M. Lu, "Histopathological image classification using novel convolutional autoencoder with residual connection," *Journal of Healthcare Engineering*, vol. 2022, p. 6496942, 2022.

[21] Y. Wang, J. Chen, and F. Zhang, "Lightweight anomaly detection for edge devices using pruning and quantization," in *IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 200–205, IEEE, 2023.

[22] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2704–2713, 2018.

[23] S. Rani and R. Kaur, "Service quality models: A literature review," *Journal of Interdisciplinary Economic Research*, vol. 5, no. 1, 2025.

[24] M. N. Almarri and E. A. Abu-Shanab, "Service quality measures: Systematic literature review and future research directions," *Journal of Internet and E-Business Studies*, vol. 2021, no. 1, pp. 1–16, 2021.

[25] J. K. Alluhaibi and K. M. Ghazi, "Design and implementation of an internet-of-things-enabled smart meter and smart plug for home-energy-management system," *Electronics*, vol. 12, no. 19, p. 4041, 2023.

[26] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Scientific Reports*, vol. 14, no. 1, p. 6086, 2024.

[27] M. Orlando, A. Estebsari, E. Pons, M. Pau, S. Quer, M. Poncino, L. Bottaccioli, and E. Patti, "A smart meter infrastructure for smart grid iot applications," *IEEE Internet of Things Journal*, vol. 9, no. 14, pp. 12529–12541, 2021.