ARTICLE

Check for updates

Journal of
Sensor Science and Technology

The Korean Sensors Society

Hybrid Deep- Learning Approach with Transformer Integration for Unknown Malware Detection in IoT Environments

Dhanya L[1,+], Chitra R[2], and Anusha Bamini A M[2]

[1]*Department of Computer Science and Engineering, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India*
[2]*Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu, India*

Cite This: *J. Sens. Sci. Technol. Vol. 34, No. 4 (2025) 289-296*

https://doi.org/10.46670/JSST.2025.34.4.289

ABSTRACT: The extensive use of Internet of Things (IoT) devices in every aspect of human life presents severe security threats due to their increased attack surfaces. The presence of unknown or novel malware is extremely difficult to identify using the conventional signature-based mechanisms. This paper proposes a hybrid malware detection framework that integrates deep learning with a real attention mechanism. An attention-enhanced autoencoder is employed to learn feature saliency and reconstruct benign behavior, with reconstruction error serving as a key anomaly indicator. The accuracy of malware detection is improved by combining the contextual dependencies of the transformer-based encoder, the cosine similarity in the autoencoder's latent space, and the anomaly scores from Isolation Forest. The individual anomaly scores from these components are integrated to capture the varied outlier behaviors. A dynamic, Receiver Operating Characteristic optimized thresholding mechanism is used to fine-tune the decision boundaries for high sensitivity and specificity. Experiments on the N-BaIoT and UNSW-NB15 dataset yielded an accuracy of 99-96% and an F1-score of 0.99-0.96 on Mirai and Bashlite malware and on DoS and back door attacks with an inference time of 13-20s. These results demonstrated the effectiveness of the framework in detecting zero-day threats with minimal memory and processing overhead.

KEYWORDS: *Internet of Things (IoT), Auto-encoder, Cosine similarity, Isolation forest, Transformer*

1. INTRODUCTION

The Internet of Things (IoT) is a modern paradigm that enhances communication and computation through the usage of sensors and devices connected through the Internet. IoT can address the various challenges in governmental, public, and private sectors through smart devices and the Internet. The diversified application of IoT includes healthcare, agriculture, logistics, retail, and several other industries.

Due to the extensive usage of IoT devices, there is an increased necessity for data security and protection systems. The reasons for various threats in IoT devices are weak passwords, a lack of encryption, and insecure communication protocols. Botnets exploit the vulnerabilities in the devices and execute large-scale attacks using the compromised devices. The differences in the security levels of various IoT devices are due to the lack of standardized security practices.

Traditional antivirus solutions, relying on signature-based methods, are primarily designed to identify known malware based on predefined signatures. Unknown malware detection technologies are essential for early threat detection and prevention, often before they can inflict significant harm. Cyber attackers continually create new malware variants to exploit vulnerabilities. Although malware detection activities are undoubtedly performed more quickly using machine-learning classification systems [3], conventional machine machine-learning techniques rely only on manually created features. Careful feature engineering and domain expertise are required to extract these properties.

Hybrid and deep learning techniques have been applied for enhancing malware detection accuracy and robustness in IoT devices. Hemalatha et al. [4] proposed a DenseNet with a data visualization technique and re weighted loss function for malware detection. Hammood et al. [5] detected malware on Android platforms using a machine-learning-based adaptive method that adapts dynamically to different attack scenarios. A multi-view, multi-kernel system with different feature inte-

[+]Corresponding author: dhanyanaren@gmail.com
Received : Jun. 26, 2025, Revised : Jul. 8, 2025, Accepted : Jul. 18, 2025

gration was emphasized by Haddad Pajouh [6] to detect malware that exhibits adversarial manipulation. Almazroi and Ayub [7] employed GhostNet ensemble using XGBoost for feature extraction and clustering with K-Means and PCA. This is combined with GRU Ensemble for malware classification.

Abdullah et al. [8] introduced the HCL-Classifier by integrating a convolutional neural network (CNN) and Long short-term memory (LSTM) models. This method detected various IoT attack types by exploiting temporal and spatial patterns in malware behavior. In a related study, Almazroi suggested a hybrid system that combines several deep- learning modules to improve the detection of malware that has not been observed before in smart IoT.

Graph Convolutional Networks(GCN) [9] detected several malware patterns from the API call sequences and generated a graph. It applied Markov chain and PCA to extract the features. from the graph for effective classification. SERLA (SEResNet50 + Bi-LSTM + Attention) [10] employed a visualization framework for malware detection based on deep neural networks. The executable files are converted to RGB images from which features are extracted and then fed to the Neural network.

The transformation of malware binaries into grayscale images for classification using transfer learning with CNNs has also been studied (DTMIC) [11]. Despite its high performance on well-known classes, this image-based approach adds preprocessing overhead. Furthermore, a number of models employed static thresholds and disregarded structural or local data relationships, which reduced the precision and adaptability.

It is a high necessity to employ a robust mechanism to detect the malignity of the data. This paper employs an attention-based-autoencoder for dynamically selecting the important features from the dataset for enhancing anomaly detection. A robust hybrid scoring mechanism incorporating the global, local, and latent-space anomaly detectors, such as Isolation Forest, cosine similarity, and transformer-based representations, is applied. In addition to this, an adaptive thresholding using receiver operating characteristic (ROC) optimization is employed for improved accuracy and generalization against novel threats with little overhead.

Our approach integrates supervised and unsupervised signals for effective prediction of known and unknown malware patterns. The proposed methodology introduces an ROC- optimized dynamic thresholding technique to adapt to evolving attack behaviors.

2. EXPERIMENTAL

The architecture of the proposed model, as shown in Fig. 1, includes the following components:

2.1 Standard Scaler

The standard scaler is a preprocessing technique that is used to normalize the features of the data set. This ensured that all features contributed equally during model training, particularly in the distance-based algorithms. Without scaling, features with larger numerical ranges dominate the learning process. This is a critical preprocessing step before feeding the data into models, such as Isolation Forest, or neural networks [12].

2.2 Attention-Enhanced Autoencoder

The attention-enhanced autoencoder learns to reconstruct benign data while focusing on the most relevant input features. A self-attention mechanism within the network allows it to assign greater weight to critical features during training [13]. When novel malware patterns deviate from the learned structure, they produce large reconstruction errors. This mechanism improves anomaly detection by adaptively identifying informative feature subsets rather than treating all features equally.

The training objective minimizes the reconstruction error between the input X and the reconstructed output $\hat{X}$, often with a regularization term for the attention weights:

$$L= \frac{1}{T}\sum_{t-1}^{T}\|x_t - \hat{x_t}\|^2 + \lambda\sum_{t=1}^{T}\alpha_t^2 \quad (1)$$

where λ is a regularization parameter to prevent overfitting to attention weights.

2.3 Cosine Similarity

Cosine similarity measures the similarity between two vectors by calculating the cosine of the angle between them [15,16]. A high value in cosine similarity metrics is the indi-

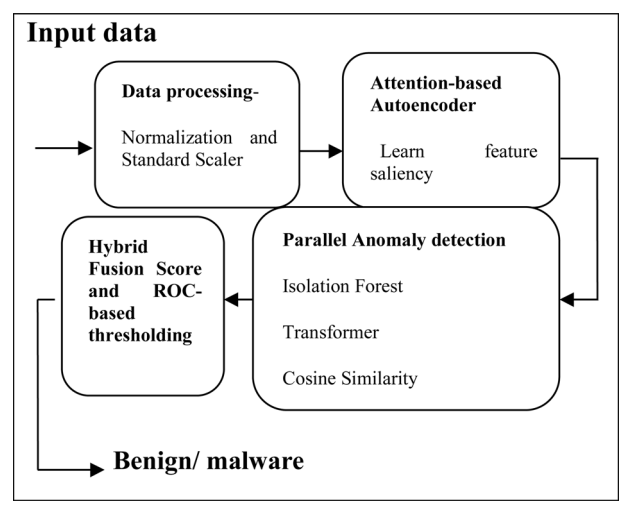

Fig. 1. Architecture of the proposed model

cator that the given input data resembles the benign behavior, and a lower metric suggests the presence of malware. This metric thus enhances malware detection without relying on labeled data samples.

2.4 Isolation Forest

Isolation Forest randomly partitions the feature space. The data points with shorter path lengths are detected as anomalies. Anomalies tend to be separated in fewer steps because of their rarity and distinctiveness. It is well -suited for high-dimensional data and does not requires no labeled examples. It provides a global anomaly score, capturing outliers that lie far from most normal samples [17-19]. For a data point x , the path length h(x) is the number of edges traversed from the root to the terminal node ina tree. The anomaly score s(x, n) for n samples is:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}} \quad (2)$$

where E(h(x)) is the average path length across multiple trees, and c(n) is the average path length of a random binary search tree with n nodes.

2.5 Transformer

A transformer [20,21] with a light weight encoder block is employed for detecting the presence of malware in tabular IoT data. A multi-head self attention mechanisms identifies the relationship between the features in the encoded latent space which in-turn enhances the detection of unknown malware patterns. This enables enhanced generalizations using more expressive modeling of complex feature interactions.

The transformer architecture is composed of stacked encoder and decoder layers, each of which is constructed using feed forward neural networks (FFNs), residual connections, multi-head self-attention, and layer normalization. An embedding layer first transforms the inputs into dense vectors, and positional encoding is used to preserve the sequence order. By computing attention weights using the query Q, key K, and value V vectors, the multi-head self-attention method enables the model to concentrate on various segments of the input sequence concurrently. This is followed by a position--wise FFN that processes each token independently. Residual connections and layer normalization are applied around both the attention and FFN sublayers to stabilize the training and enable deeper architectures, as shown in Fig. 2. By the computation of attention weights utilizing the query Q, key K, and value V vectors, the multi-head self-attention method enables

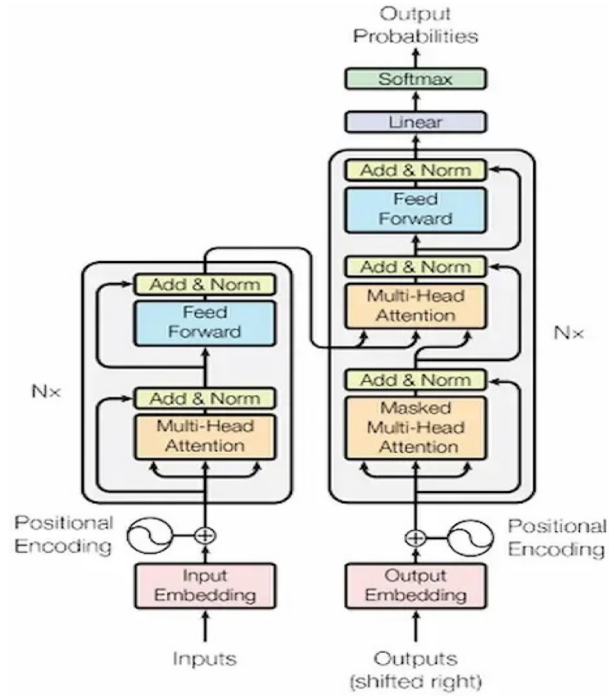

Fig. 2. Basic transformer architecture

the model to concentrate on various segments of the input sequence concurrently. This is followed by a position-wise FFN that processes each token independently. Residual connections and layer normalization are applied around both the attention and FFN sublayers to stabilize training and enable deeper architectures as shown in Fig 2.

A transformer encoder block was incorporated to model the complex temporal and contextual dependencies within the network behavior sequences. The encoder is composed of of N identical layers , each of which includes a multi-head self-attention sub-layer, where queries Q , keys K, and values V are linearly transformed, and attention is computed as $(\frac{QK^T}{\sqrt{d_K}})$ V to focus on the pertinent input parts. In the decoder, this structure is mirrored by N layers, followed by masked multi head self attention, followed by a linear layer and softmax for output creation, masked multi head self attention, and an encoder attention sub-layer to integrate encoder outputs.

Given an input sequence $X=[x_1,x_2,\ldots x_T]$ where each $x_t \in R^d$ is a feature vector at time step t, the transformer applies multi-head attention:

$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_K}})\ V \quad (3)$$

For each attention head $h \in \{1,\ldots H\}$ where W is the weight:

$$Q_h = XW_h^Q \quad (4)$$

$$K_h = XW_h^K \quad (5)$$

$$V_h = XW_h^V \quad (6)$$

$$\text{Head}_h = \text{Attention}(Q_h, K_h, V_h) \quad (7)$$

The outputs from all heads are concatenated and linearly projected.

The multihead attention layer is followed by a feedforward layer with residual connections and layer normalization, enabling the model to learn contextual representations and subtle sequential deviations in the network behavior. The transformer encoder outputs a refined latent representation Z_{trans} which contributes to an anomaly score based on the divergence from the benign behavior, modeled using the Mahalanobis distance with known benign embeddings.

2.6 Hybrid Score Fusion

The presence of anomaly in the data is detected by the hybrid score fusion, which integrates the outputs of Isolation Forest, cosine similarity, and transformer. The detection is based on the global interactions, feature reconstruction, and local context of the features. Anomalies are identified based on the normalized and weighted fusion scores. Thus the ensemble promotes robustness with reduced false positives when compared to single-component detectors.

The final anomaly decision is based on a combination of complementary scores rather than a single indicator. The reconstruction loss of the auto-encoder is given by:

$$S_1(x) = \left\| x - \hat{x} \right\|^2 \quad (8)$$

The cosine similarity in latent space is:

$$S_2(x) = 1\text{-}\cos(z_x, \mu_{benign}) \quad (9)$$

where z_x is the latent code of input x and μ_{benign} is the centroid of the benign.

The Isolation Forest Score score $S_3(x)$, is based on the average path length in the decision trees.

The transformer-based semantic Deviation deviation score $S_4(x)$, is modeled using the Mahalanobis distance. These scores are first normalized and then combined using a learned or adaptive weighted ensemble.

$$S_{final}(x) = \sum_{i=1}^{4} w_i S_i(x) \quad \text{where} \quad \sum w_i = 1 \quad (10)$$

2.7 ROC - based Adaptive Thresholding

The detection accuracy of malware is enhanced by employing an optimal cutoff based on the ROC curve without relying on static threshold values. Moreover, the model can adapt to the dynamic distribution of data in the test dataset, which in turn enhances the specificity and sensitivity of novel malware. The selected dynamic threshold value decides whether the samples are benign or malicious.

3. RESULTS AND DISCUSSIONS

The experiment is conducted using Python 3 and a Jupyter Notebook. The dataset used is NBaIoT [22], which includes malware affecting IoT devices, namely, Mirai and Bashlite. We trained the model using benign data with 49,000 samples. The test data are a combination of benign and malware (Mirai/Bashlite) samples. The dataset contains 115 features, which represents network-level parameters. In our implementation, the transformer encoder consists of four layers, each with four attention heads, an embedding dimension of 128, a feedforward dimension of 512, and a dropout rate of 0.1.

Fig. 3 shows the input features prioritized by the attention-enhanced autoencoder during encoding. Using a self-attention mechanism in the input layer, the model assigns weights to each feature based on its importance in accurately reconstructing benign behavior. Features with higher attention weights are considered more significant for capturing the structure of normal data and, improving both the model's anomaly detection efficiency and interpretability by identifying key features for decision-making.

A confusion matrix is a table used to evaluate the performance of a classification model by showing the number of correct and incorrect predictions broken down by each class [23]. Fig. 4 shows the confusion matrix obtained from the Mirai dataset. The proposed approach had a True positive rate of 7424 samples and a true negative of 7192 samples on the Mirai

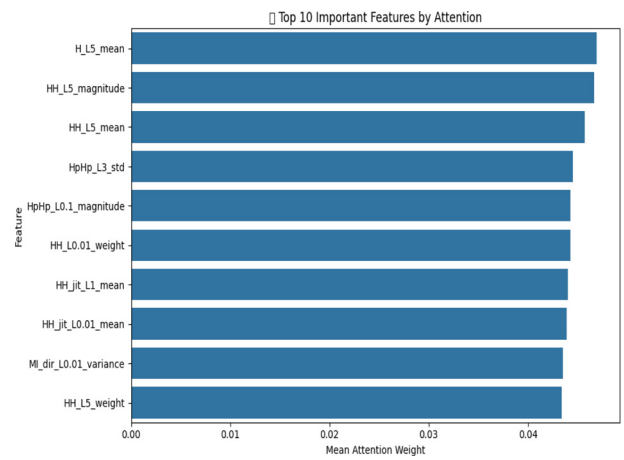

Fig. 3. Top10 features selected by the autoencoder

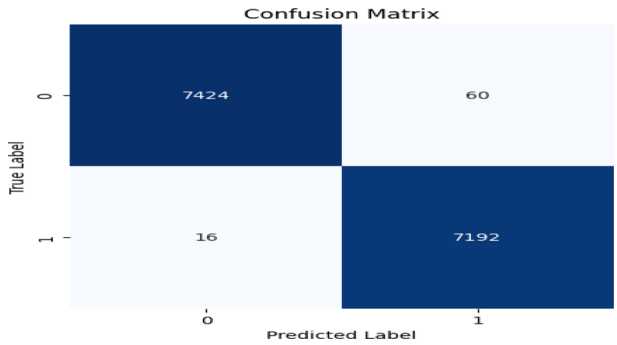

Fig. 4. Confusion matrix for the Mirai dataset

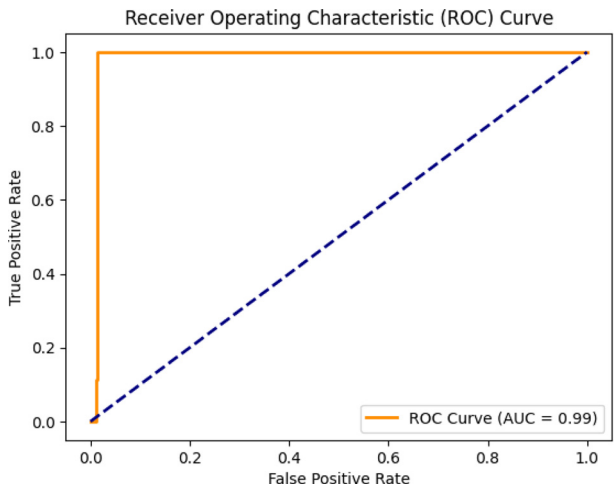

Fig. 5. ROC Curve on the Mirai dataset

dataset. This reflects the efficiency of the approach for detecting normal and abnormal instances.

ROC and precision–recall curves are the two important evaluation metrics used to assess the performance of binary classification models. The capacity of the model to distinguish between positive and negative classes by altering the classification threshold is graphically shown by the ROC curve in Fig. 5. Plotting the TPR versus FPR at various threshold settings yields the required results. The top-left corner of the plot is the point where the ROC curve of a perfect classifier passes (TPR = 1, FPR = 0), suggesting high sensitivity (accurately predicting positives) and a low FPR. An area under the ROC curve (AUC-ROC) value closer to 1 indicates a better-performing mode.

The precision-recall curve [24] is another metric for evaluating the classification models, particularly with imbalanced datasets. The precision indicates the correct positive predictions out of the total positive predictions, whereas recall is the measure of actual positive instances identified by the model. The precision-recall curve is generated by altering the model's decision threshold and monitoring the changes in precision and

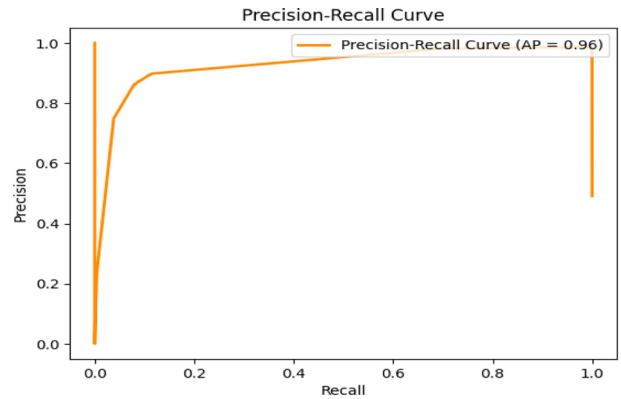

Fig. 6. Precision-recall curve on the Mirai dataset

Table 1. Performance Evaluation on the Mirai and Bashlite dataset.

Dataset	Accuracy	Precision	Recall	F1-Score
Mirai	0.9938	1.00	0.99	0.99
Bashlite	0.9843	0.99	1.00	0.99

Table 2. Performance Evaluation on the UNSW-NB15 dataset.

Dataset	Accuracy	Precision	Recall	F1-Score
DoS	0.982	0.99	0.99	0.99
Backdoor	0.964	0.97	0.97	0.97

recall at each stage. A high value of the precision-recall curve that crosses the top-right corner (with precision = 1, recall = 1) is an indicator of a perfect classifier. Fig. 6 shows the performance highlights.

The proposed method produced significant results in terms of accuracy, F1-score, and precision as listed in Table 1. The false-positive and false-negative values are significantly lower than the true- positives and true-negative values, as per the classification report.

The experiment is conducted on the UNSW-NB15 dataset [25], which contains benign and malware data, such as DoS and backdoor attacks. The model is trained on normal data and tested on the malware samples; and its performance is highlighted ion Table 2.

In the context of outlier detection algorithms, outlier scores are assigned to the data points in the dataset. Outlier scores are numerical values that indicate the degree to which each data point in the sample deviates from the normal pattern. A higher value of outlier score indicates the data points are anomalous, which deviates from the benign data points. Fig. 7 shows the outlier score distributions for the datasets.

The time taken for detection by the proposed method and the memory usage on the two datasets are listed is given in Table 3. The proposed approach is lightweight with a faster detection time, and, hence, suitable for real- time IoT scenarios.

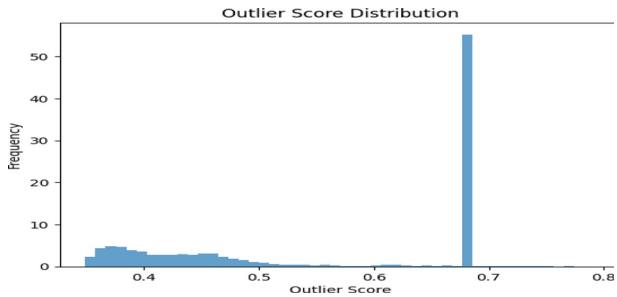

Fig. 7. Outlier score distribution of the datasets

Table 3. Time and Memory usage on Mirai-Bashlite datasets.

Dataset	Time of detection	Memory usage
Mirai	13.26 s	36 kilobyte
Bashlite	20.38 s	36 kilobyte

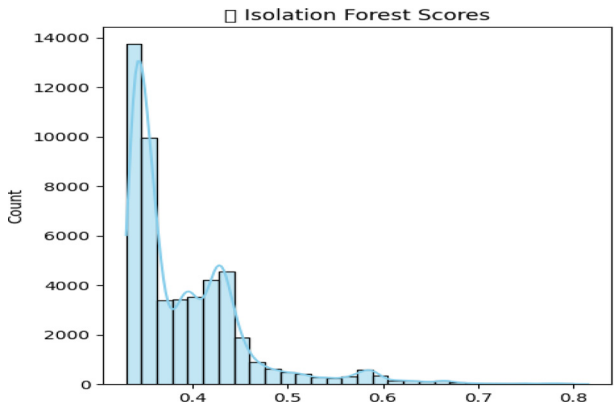

Fig. 8. Isolation Forest scores from the datasets

Isolation Forest (IF) scores quantify how easily a data point can be isolated using random feature splits. The model builds multiple decision trees, and isolated samples with fewer splits are likely to be anomalies. In this context, the scores are inverted so that higher values indicated a greater likelihood of being anomalous. IF is particularly effective at identifying global anomalies, that is, —data points that are significantly different from the overall distribution, as shown in Fig. 8.

The contextual dependencies between the features are modeled by passing each encoded feature vector through a transformer block. The deviations between the original and transformed embeddings determine the anomaly score. The higher values of normalized scores indicate the anomalous behavior. Transformer-based scoring excels in identifying complex, context-aware anomalies particularly those involving feature interactions that do not stand out in isolation. Fig. 9 shows the transformer-based scores.

In Fig. 10, normal data samples exhibit low cosine distances are aligned to the normal behavioral patterns in the training dataset, whereas the anomalous samples have higher cosine

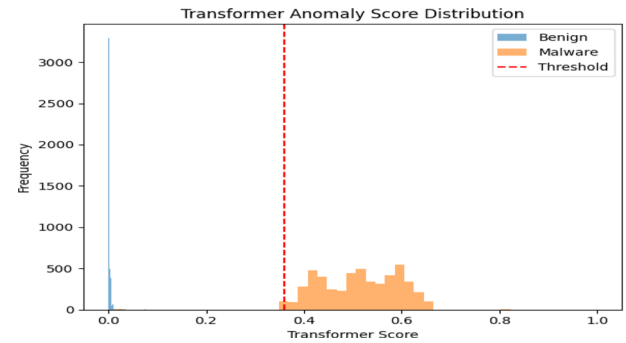

Fig. 9. Transformer based scores on the datasets

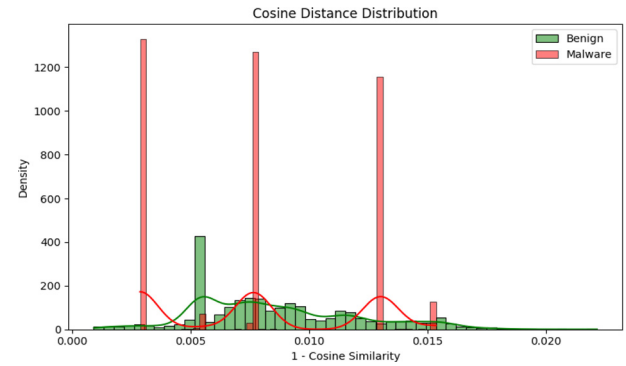

Fig. 10. Cosine similarity scores of the datasets

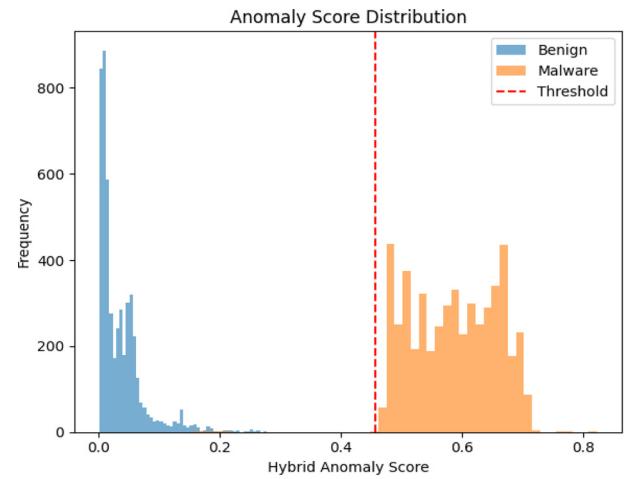

Fig. 11. Anomaly detection plot for the Mirai dataset

distances. Thus, the application of cosine similarity complemented with the reconstruction error accelerates the process of discriminating normal and malicious patterns.

The anomaly detection plot displays data instances along the X –axis and the anomaly score along the Y-axis as shown in Fig 11. The plot allowed to observe the distribution of the anomaly scores and identify instances that deviated significantly from the normal instances. Instances with anomaly scores beyond the threshold are more likely to be considered anomalies. Color-coding helps differentiate between normal

Table 4. Performance Analysis of Individual Components

Method	Accuracy	F1-Score	Precision
Attention-based Autoencoder (AE)	0.951	0.94	0.92
AE + Latent Space Cosine Similarity	0.963	0.96	0.94
AE + Isolation Forest	0.96	0.95	0.93
AE + Transformer Encoder	0.968	0.96	0.95
AE + All Three Modules (Cosine + IF + Transformer)	0.995	0.99	0.99

Table 5. Comparison with Other State-of-the-Art Models

Method	Accuracy	F1-Score	Precision
LSTM-Auto-encoders and Multilayer Perceptron (Dataset- UNSW-NB15)	99.6%	0.997	0.996
1D Conv-Network (Dataset- CIC IoT 2023)	98.36%	0.99	1.00
BERT-based Feed Forward Neural Network Framework (Dataset- IOT2-CTU/IoT 23)	98%	0.979	0.99
FedPCA Dataset- UNSW-NB15	82.15%	0.875	0.810
Proposed method	99.3%	0.99	1.00

and anomalous instances.

The accuracy, F1-score and precision obtained using the individual components such as the autoencoder, autoencoder with latent cosine similarity, autoencoder with Isolation Forest , autoencoder with transformer are listed in Table 4.

3.1. Comparison with State-of-Art models:

Table 5 depicts how the ensemble mode outperforms various state-of-the-art models, such as LSTM-Autoencoders and Multilayer Perceptron, 1D Conv Network, BEFSONET, FedPCA in terms of accuracy, F1-score, and precision on datasets, such as UNSW-NB15, CIC IoT 2023 and IOT2-CTU/IoT 23 which have normal and malware samples of IoT data as csv files.

4. CONCLUSIONS

This paper introduced a hybrid anomaly-detection system that combines a transformer, a cosine-based analysis, and an attention-enhanced autoencoder with Isolation Forest. The autoencoder learns to reconstruct benign IoT behaviors, whereas the attention highlights the most salient features. Combined with ensemble-based anomaly scoring and ROC-optimized thresholding, the system effectively detected unknown malware. Experimental results on the Mirai, Bashlite, DoS, and back door datasets demonstrated high accuracy, robust generalization, and efficient computation, making it well -suited for real-world IoT and edge environments. In the future, the model should be fine- tuned and applied on more datasets.

CRediT Authorship Contribution Statement

Dhanya L: Conceptualization, Methodology, Writing of the original draft. **Chitra R:** Supervision and validation. **Anusha Bamini A M:** Writing – editing.

Declaration of Competing Interest

The authors declare that they have no competing financial interests or personal relationships that may have influenced the work reported in this study.

Acknowledgements

No external funding was received for this study.

REFERENCES

[1] S. Kumar, P. Tiwari, M. Zymbler, Internet of things is a revolutionary approach for future technology enhancement: A review, J. Big Data 6 (2019) 1–21.
[2] A.H. Hussein, Internet of things (IOT): Research challenges and future applications, Int. J. Adv. Comput. Sci. Appl. 10 (2019) 77–83.
[3] M. Wazid, A.K. Das, J.J. Rodrigues, S. Shetty, Y. Park, IoMT malware detection approaches: Analysis and research challenges, IEEE Access 7 (2019) 182459–182476.
[4] J. Hemalatha, S.A. Roseline, S. Geetha, S. Kadry, R. Damaševičius, An efcient densenet-based deep learning model for malware detection, Entropy 23 (2021) 344.
[5] L. Hammood, İ.A. Doğru, K. Kılıç, Machine learning-based adaptive genetic algorithm for android malware detection in autodriving vehicles, Appl. Sci. 13 (2023) 5403.
[6] H. HaddadPajouh, An Adversarially Robust Multi-view Multi-kernel Framework for IoT Malware Threat Hunting, Ph.D. Qualifying Exam Report, University of Guelph, Guelph, Canada, 2021.
[7] A.A. Almazroi, N. Ayub, Enhancing smart IoT malware detection: A GhostNet-based hybrid approach, Systems 11 (2023) 547.
[8] M.A. Abdullah, Y. Yu, K. Adu, Y. Imrana, X. Wang, J. Cai, HCL-Classifier: CNN and LSTM based hybrid malware classifier for Internet of Things (IoT), Future Gener. Comput. Syst. 142 (2023) 41–58.
[9] S. Li, Q. Zhou, R. Zhou, Q. Lv, Intelligent malware detection based on graph convolutional network, J. Supercomput. 78 (2022) 4182–4198.
[10] Y. Jian, H. Kuang, C. Ren, Z. Ma, H. Wang, A novel framework for image-based malware detection with a deep neural network, Comput. Secur. 109 (2021) 102400.
[11] S. Kumar, B. Janet, DTMIC: Deep transfer learning for

malware image classification, J. Inf. Secur. Appl. 64 (2022) 103063.
[12] Devashi Gulati, Feature Selection and Analysis in Machine Learning and Data Science. https://medium.com/couture-ai/feature-analysis-for-feature-selection-in-machine-learning-and-data-science-f3e5a3e4c571, 2019 (Accessed 5 June 2025).
[13] Z. Niu, G. Zhong, H. Yu, A review on the attention mechanism of deep learning, Neurocomputing 452 (2021) 48–62.
[14] G. Brauwers, F. Frasincar, A general survey on attention mechanisms in deep learning, IEEE Trans. Knowl. Data Eng. 35 (2021) 3279–3298.
[15] T.P. Rinjeni, A. Indriawan, N. A. Rakhmawati, Matching Scientific Article Titles using Cosine Similarity and Jaccard Similarity Algorithm, Procedia Comput. Sci. 234 (2024) 553–560.
[16] S. Mukherjee, R. Sonal, A reconciliation between cosine similarity and Euclidean distance in individual decision-making problems, Indian Econ. Rev. 58 (2023) 427–431.
[17] P. Zhang, F. He, H. Zhang, J. Hu, X. Huang, J. Wang, et al., Real-time malicious traffic detection with online isolation forest over SD-WAN, IEEE Trans. Inf. Forensics Secur. 18 (2023) 2076–2090.
[18] C. Mougan, Isolation forest from Scratch. https://medium.com/data-science/isolation-forest-from-scratch-e7e5978e6f4c, 2020 (Accessed 5 June 2025).
[19] A.M.A. Fadul, Anomaly Detection based on Isolation Forest and Local Outlier Factor, Master's Thesis, Africa University, Mutare, Zimbabwe, 25 th May 2023.
[20] T. Lin, Y. Wang, X. Liu, X. Qiu, A survey of transformers, AI Open 3 (2022) 111–132.
[21] X. Amatriain, A. Sankar, J. Bing, P.K. Bodigutla, T.J. Hazen, M. Kazi, Transformer models: an introduction and catalog, arXiv., https://arxiv.org/abs/2302.07730 (2023).
[22] K. Naveed, N-baiot dataset to detect IoT Botnet attacks. https://www.kaggle.com/datasets/mkashifn/nbaiot-dataset, 2020 (Accessed 5 June 2025).
[23] J. Erbani, P.-É. Portier, E. Egyed-Zsigmond, D. Nurbakova, Confusion Matrices: A Unified Theory, IEEE Access 12 (2024) 181372–181419.
[24] J. Miao, W. Zhu, Precision–recall curve (PRC) classification trees, Evol. Intel. 15 (2022) 1545–1569.
[25] UNSW, The UNSW-NB15 Dataset. https://research.unsw.edu.au/projects/unsw-nb15-dataset, 2021 (Accessed 5 June 2025).