Fusion-Driven Spatio-Temporal Deep Learning for IoT Intrusion Detection: Integrating CNN, TCN, and CapsuleNet Features with LSTM

B. Chempavathy
Department of Computer Science and Engineering, School of Engineering, Dayananda Sagar University, Bangalore, Karnataka, India
chempa.tusti@gmail.com (corresponding author)

S. K. Mouleeswaran
Department of Computer Science and Engineering, School of Engineering, Dayananda Sagar University, Bangalore, Karnataka, India
mouleeswaran-cse@dsu.edu.in

Received: 10 June 2025 | Revised: 23 July 2025 and 13 August 2025 | Accepted: 20 August 2025

 DOI: https://doi.org/10.48084/etasr.12662

ABSTRACT

The rapid growth of Internet of Things (IoT) devices has increased their susceptibility to various cyber-attacks, making efficient intrusion detection crucial. This study presents a hybrid deep learning model that integrates a Convolutional Neural Network (CNN), a Temporal Convolutional Network (TCN), a Capsule Network (CapsuleNet), and a Long Short-Term Memory (LSTM) to examine complex network behavior patterns for IoT security. This architecture uses CNN and TCN for spatial and temporal feature extraction, CapsuleNet to augment representational capacity, and LSTM to describe long-term dependencies. The model is trained and validated using the NSL-KDD dataset, which encompasses five classes: Normal, Remote to Local (R2L), User to Root (U2R), Denial of Service (DoS), and Probe. Data augmentation using SMOTE reduces class imbalance, while performance measures such as accuracy, recall, and F1 score provide a comprehensive evaluation of performance. The proposed hybrid model achieved a maximum accuracy of 98.71%, outperforming individual models that reached accuracies from 91.02% to 96.76%. This integration of approaches provides an effective and dependable solution for IoT intrusion detection, demonstrating its efficacy in protecting dynamic low-power IoT settings.

Keywords-IoT security; intrusion detection system; feature fusion; spatio-temporal deep learning; intelligent threat detection

I. INTRODUCTION

The IoT has revolutionized our interaction with technology, enabling seamless connection and automation across several industries. The capacity to distinguish between normal and malicious traffic has been examined using several ML models. In [1], a novel anomaly-based IDS, using Machine Learning (ML) and Deep Learning models, was discussed. In [2], a Deep Reinforcement Learning (DRL) architecture used network data to identify and categorize attacks. In [3], a conditional tabular generative adversarial network was used to address the data imbalance problem. In [4], a system based on the Rabin-Karp algorithm was discussed to detect abnormal nodes in IoT networks. In [5], a hybrid ML had a high computational cost and was sensitive to scaling, resulting in loss of information. To effectively detect intrusions in ever-changing IoT contexts, an influential Synaptic Intelligent Convolutional Neural Network (SICNN) was discussed in [6]. Despite the need for device-specific protocols, in [7], the proliferation of IoT security breaches led to the creation of an IDS using Support Vector Machine (SVM). A comprehensive testing application for IDS was described in [8], utilizing the Zed Attack Proxy to protect web applications in IoT devices. In [9], a Generative Adversarial Network (GAN)-based IDS was described, which identifies issues that affect the Message Queueing Telemetry Transfer (MQTT) protocol. In [10], a detailed analysis of the threats facing IoT networks was provided. Different ML methods were examined in [11-14], using different feature selection and ensemble methods. In [15], mutual information-based feature selection for IDS was used, along with fuzzy Gaussian functions, and classification was performed using CNN, LSTM, and a Deep Belief Network (DBN). In [16], a Bayesian decision with fuzzy logic was employed for intrusion detection. In [17], an efficient IDS integrated DL models. In [18-19], various ML methods were used for intrusion detection in IoT.

The key contributions of this study are:

- Presents a DL architecture that integrates CNN, TCN, CapsuleNet, and LSTM to effectively extract and integrate spatial and temporal features for intrusion detection.
- Demonstrates superior performance by using the synergistic advantages of each constituent network.
- Validates the proposed model using the NSL-KDD dataset for five intrusion classes.
- Provides a scalable and flexible DL framework that is appropriate for real-time IoT security applications, including implementation on edge and resource-limited devices.

These contributions enhance the accuracy of the proposed IDS, which overcomes the inadequate temporal modeling and suboptimal feature representation of existing techniques.

II. PROPOSED ARCHITECTURE FOR INTRUSION DETECTION

The proposed system is a hybrid DL framework aimed at improving the detection of intrusions into the IoT through the integration of spatial and temporal features. It integrates several DL models to enhance accuracy, resilience, and flexibility in dynamic network situations. The proposed system enables the identification of both spatial and temporal patterns in complex network data. CNN, TCN, and CapsuleNet extract diverse spatial features, while LSTM captures temporal dependencies. Figure 1 shows the integration of CNN, TCN, and CapsuleNet to extract features, which LSTM processes for accurate IoT intrusion detection.

A. Data Preprocessing

Data preprocessing steps include handling missing values, removing duplicates, standardizing numerical attributes, encoding categorical variables, and identifying the most important aspects. Figure 2 shows the data-cleaning process diagram.

The next step is to divide the preprocessed data into sequences to enable temporal analysis. Interpolation is used for minor gaps, and entries with significant missing data are eliminated. Data normalization is a preprocessing procedure that adjusts numerical features to a uniform range, ensuring that all characteristics contribute equally to the model. This is especially crucial for DL models since it enhances convergence speed and overall performance. Normalization was performed using:

$$x_{normalized} = \frac{x-\mu}{\sigma} \quad (1)$$

where x is the original value, μ is the feature mean, and σ is the standard deviation of the features. The encoding process transforms categorical or non-numeric input into a numerical representation that is comprehensible to ML algorithms.

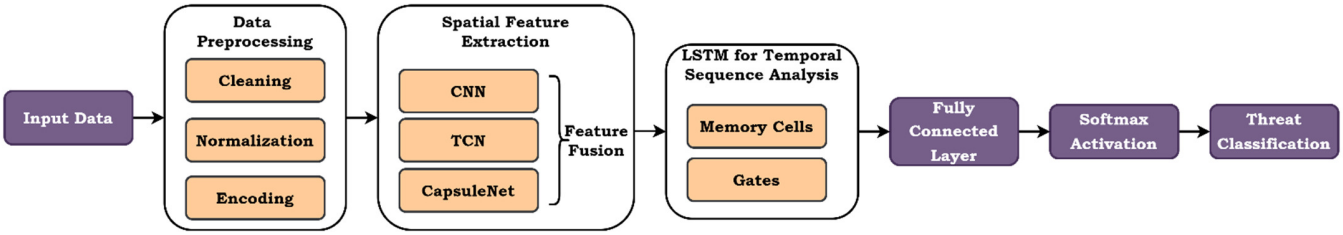


Fig. 1. Block diagram of CNN-TCN-CapsuleNet feature extraction with LSTM-based classification.

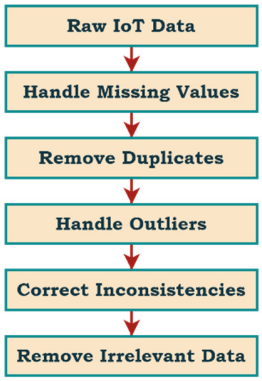


Fig. 2. Data cleaning process.

B. Spatial Feature Extraction

The CNN module uses 64 filters of size 1×3 to extract spatial characteristics along the sequence of IoT data. The convolution at each layer is given by:

$$h^{(l)} = f(W^{(l)} * h^{(l-1)} + b^{(l)}) \quad (2)$$

where $W^{(l)}$ are 1×3 convolution kernels, $h^{(l-1)}$ is the input from the previous layer, $b^{(l)}$ is the bias, and f is ReLU activation. This narrow kernel effectively captures localized sequential patterns while maintaining temporal order.

TCN uses 128 filters with a 1×3 kernel and dilated causal convolutions to capture temporal and spatial correlations.

$$y(s) = \sum_{i=0}^{2} f(i) \cdot x_{s-d \cdot i} \quad (3)$$

where kernel size $k = 3$ corresponds to a filter length of 1×3, the dilation factor d regulates the spacing, $f(i)$ represents the filter weights, and input x is sampled using dilation to capture long-range dependencies. The residual connections add stability as:

$$o = Activation\big(x + ConvBlock(x)\big) \quad (4)$$

CapsuleNet uses capsules with a dimension of 256 to encapsulate spatial hierarchies. It handles features modified by learned matrices via dynamic routing:

$$s_j = \sum_i c_{ij} W_{i,j} u_i \quad (5)$$

with squashing:

$$v_j = \frac{||s_j||^2}{1+||s_j||^2}\frac{s_j}{||s_j||} \quad (6)$$

The 1×3 filters maintain feature continuity across sequences, whereas capsules preserve hierarchical connections within sequences. Figure 3 shows the overall architecture of the proposed hybrid model for IoT intrusion detection.

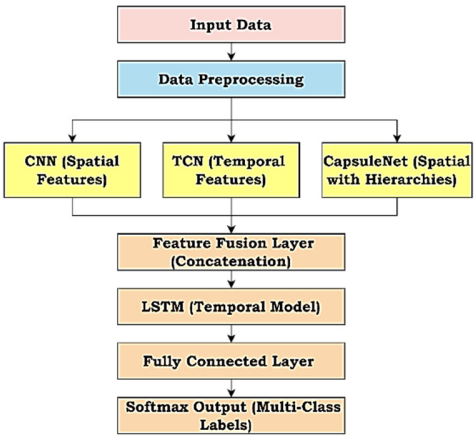


Fig. 3. Fusion architecture integrating CNN, TCN, and CapsuleNet.

Each model handles the same pre-processed input data in parallel. CNN extracts local spatial features. TCN captures sequential dependencies using temporal convolutions. CapsuleNet maintains spatial hierarchy and poses connections. The fusion layer integrates outputs from CNN, TCN, and CapsuleNet, generating a more comprehensive feature set for accurate classification using a concatenation technique. The LSTM mechanism uses its integrated characteristics to represent temporal interdependence. The fully connected and softmax output layer converts learned features into probabilities for multi-class classification.

C. Temporal Sequence Analysis with LSTM

The LSTM network is a type of Recurrent Neural Network (RNN) designed for analyzing sequential data by capturing temporal relationships. The proposed method utilizes LSTM to analyze the spatial characteristics derived from the CNN, making it easier to identify trends and outliers in IoT data collected over time. The limitations of traditional RNNs, such as the vanishing gradient problem that prevents the development of long-term associations, are specifically addressed in the design of LSTM. The data that is to be erased from the memory cell is defined by the forget gate f_t, which can be expressed as:

$$f_t = \sigma(W_f \cdot [h_{t-1,x_t} + b_f] \quad (7)$$

The weight matrix W_f, the hidden state h_{t-1} from the previous time step, the input x_t from the current time step, and the bias term b_f make up the forget gate. The fresh data to be stored in the memory cell is identified by the input gate i_t.

$$i_t = \sigma(W_i \cdot [h_{t-1,x_t} + b_i) \quad (8)$$

$$\hat{C}_t = \tanh(W_C \cdot [h_{t-1,x_t} + b_C) \quad (9)$$

where W_i is the weight matrix for the input gate, W_C is the weight matrix for the candidate memory cell, $\hat{C}_t$ is the candidate memory cell state, and tanh is the hyperbolic tangent activation function. Figure 4 represents an LSTM diagram for sequential processing, using memory cells and gates to analyze temporal dependencies for IoT threat classification.

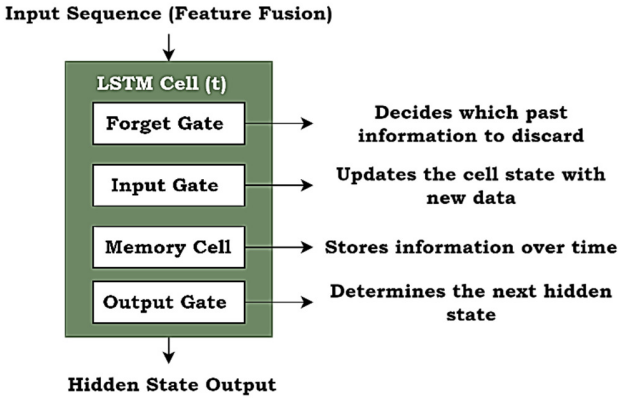


Fig. 4. LSTM architecture for sequential threat detection in IoT.

LSTMs control the flow of data through a memory cell and three gates: the input, forget, and output gates. The memory cell update C_t modifies the memory cell state by integrating the prior state with the new candidate state given by:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t \quad (10)$$

The output gate is defined as:

$$C_t = \sigma(W_o \cdot [h_{c-1,x_t} + b_o) \quad (11)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (12)$$

In this case, the weight matrix W_o represents the output gate, and h_t denotes the hidden state at the present time step.

D. Intrusion Classification

The fusion architecture integrates spatial information from CNN, TCN, and CapsuleNet into a unified feature vector y_t at each time step t:

$$y_t = softmax(W_y \cdot h_t + b_y) \quad (13)$$

where

$$h_t = LSTM\left(Concat\left(x_t^{CNN}, x_t^{TCN}, x_t^{CapsuleNet}\right)\right) \quad (14)$$

The integrated features from CNN, TCN, and CapsuleNet are input into the LSTM to generate the hidden state h_t, which is further processed via a softmax layer for classification. The softmax activation produces probability ratings for each category as:

$$P(y = c) = \frac{e^{z_c}}{\sum_{k=1}^{C} e^{z_k}} \quad (15)$$

III. RESULTS AND DISCUSSION

The NSL-KDD dataset [20] includes five classes, namely normal, R2L, U2R, DoS, and Probing, and was utilized to evaluate the proposed IDS. NSL-KDD was selected for its minimal redundancy and widespread use in evaluating IDSs. Table I presents the original distribution of samples for each NSL-KDD attack class.

TABLE I. ORIGINAL NSL-KDD DATASET DISTRIBUTION BEFORE APPLYING SMOTE AUGMENTATION

Attack class	Train	Test	Total
Normal	67343	9711	77,054
DoS	45927	7458	53,385
Probe	11656	2421	14,077
R2L	995	2754	3,749
U2R	52	200	252

To address the class imbalance, data augmentation was implemented using the Synthetic Minority Over-sampling Technique (SMOTE) [21]. This method synthetically generates new samples for minority classes, such as R2L and U2R, reaching 5,000 samples. With 5,000 samples analyzed in each class, the dataset was split for training (60%), validation (20%), and testing (20%). This distribution provides an efficient performance of the CNN+TCN+CapsuleNet-LSTM model in identifying diverse attack types, while preserving generalization against unencountered threats in IoT security.

A dropout of 0.3 is also employed, which randomly deactivates neurons during training, mitigating overfitting by promoting robust feature learning and enhancing generalization to new data. The execution time of the proposed model is approximately 280 seconds for training and 0.023 seconds for inference per sample, making it suitable for cloud-based intrusion detection rather than real-time deployment in resource-limited IoT environments. All models are trained from scratch using the NSL-KDD dataset without pretrained weights. The model's performance was evaluated using accuracy, precision, recall, and F1-score.

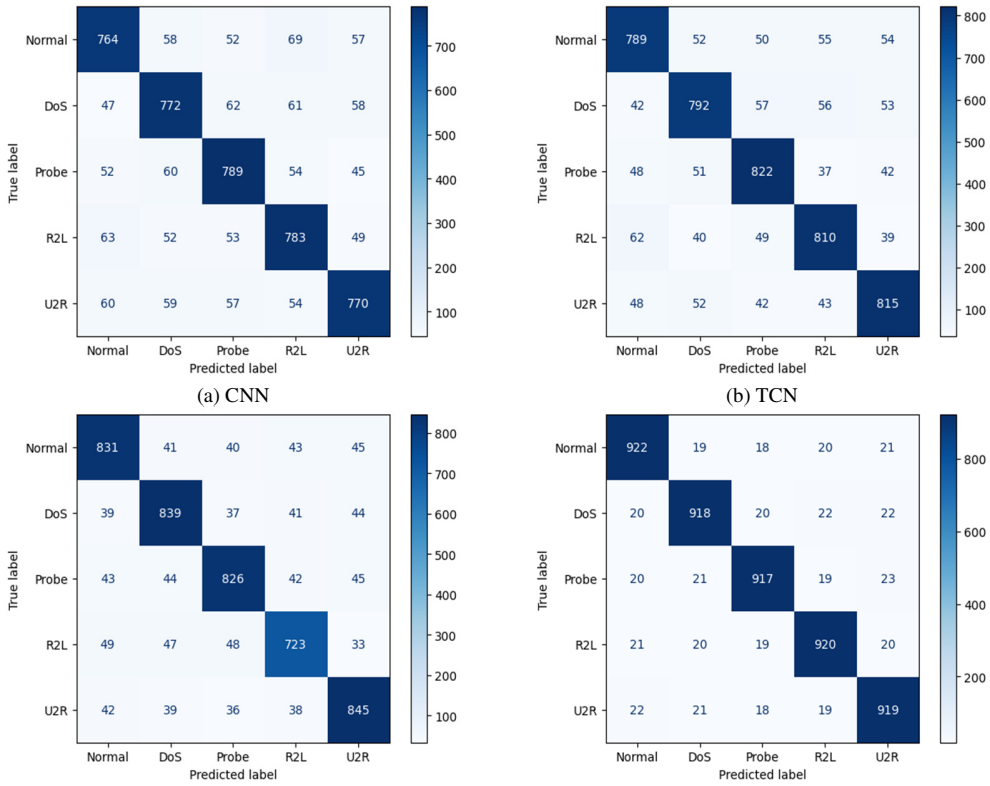
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (16)$$

$$Precision = \frac{TP}{TP+FP} \quad (17)$$

$$Recall = \frac{FP}{TP+FN} \quad (18)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (19)$$

Figure 5 shows the confusion matrices for CNN, TCN, CapsuleNet, and LSTM on the NSL-KDD Dataset, demonstrating differing degrees of classification accuracy. CNN achieved 91.02%, exhibiting significant misclassifications across all classes. TCN enhanced its performance to 92.22%, demonstrating superior management of DoS and Probe assaults. CapsuleNet achieved 93.31%, demonstrating more equitable detection, particularly in the U2R and R2L classes. LSTM achieved superior performance, with an accuracy of 96.76%, exhibiting robust temporal learning and minimal misclassification across classes.



(a) CNN (b) TCN

(c) CapsuleNet (d) LSTM

Fig. 5. Confusion matrix comparison of CNN, TCN, CapsuleNet, and LSTM models.

Figure 6 shows that the proposed fusion model achieved good accuracy across all intrusion classes, with few misclassifications.

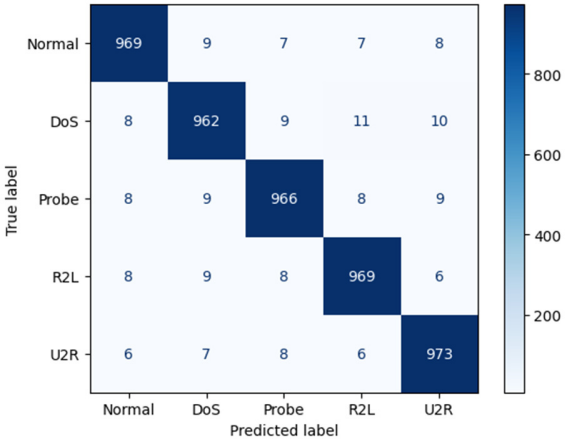


Fig. 6. Confusion matrix of the proposed CNN+TCN+CapsuleNet-LSTM intrusion detection model.

The accuracy findings demonstrate that the integration of several DL models substantially improves intrusion detection efficacy. The individual accuracy of CNN, TCN, CapsuleNet, and LSTM ranged from 91.02% to 96.76%, but the proposed fusion model, which integrates their strengths, achieves 98.71% accuracy. This enhancement demonstrates the effectiveness of combining spatial and temporal information for more resilient and accurate detection of various IoT network threats. The fusion method reduces misclassification and enhances overall system dependability. Table II shows the hyperparameters used for each component of the fusion model.

TABLE II. PARAMETER SETTINGS OF THE PROPOSED FUSION MODEL

Parameter	Value
CNN filter size	1×3
CNN number of filters	64
TCN filter size	1x3
TCN number of filters	128
CapsuleNet capsules	256
Activation function - Input	ReLU (CNN, TCN), Squash (CapsuleNet)
Activation function – Output	Sigmoid
LSTM hidden units	128
LSTM activation	tanh
Dropout rate	0.2
Optimizer	Adam
Learning rate	0.001
Batch size	64
Epochs	10

Figure 7 shows the training and validation loss of the proposed CNN+TCN+CapsuleNet-LSTM model over several epochs. Training loss begins at 0.7 and steadily decreases to below 0.1 by the 10th epoch. The validation loss starts slightly higher than the training loss, but steadily decreases over time, reaching below 0.2.

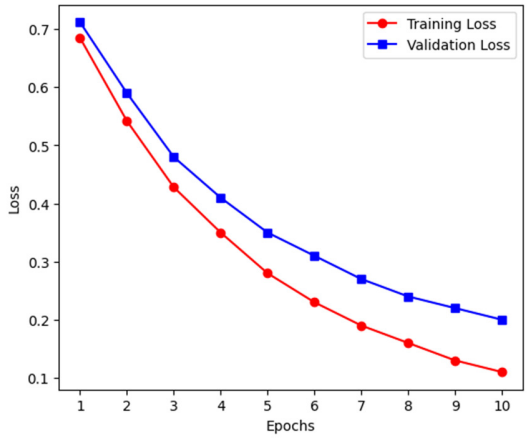


Fig. 7. CNN+TCN+CapsuleNet-LSTM loss curve.

Figure 8 shows a comparison of precision, recall, and F1-score for each intrusion class, highlighting the consistently superior performance of the proposed fusion model across all classes, which signifies robust and equitable detection capabilities. This graph demonstrates the superior and consistent performance of the proposed CNN+TCN+CapsuleNet-LSTM model across all classes. The precision, recall, and F1-score metrics are typically above 95%, indicating efficient detection with low false positives and negatives. The little discrepancies across classes signify intrinsic changes in attack difficulty, although the aggregate measures demonstrate strong classification proficiency.

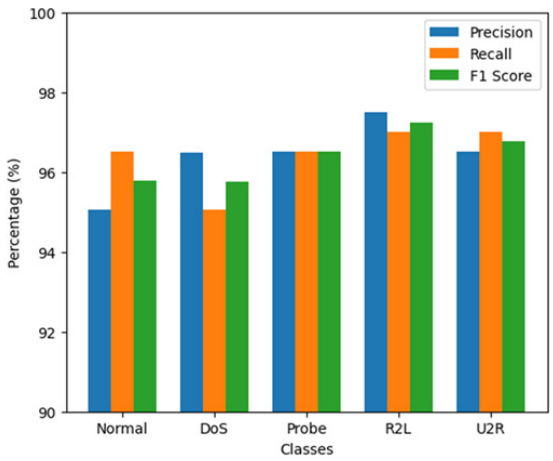


Fig. 8. Performance metrics of the proposed CNN+TCN+CapsuleNet-LSTM model.

Figure 9 shows the improved performance of the proposed CNN+TCN+CapsuleNet-LSTM model compared to traditional techniques [22], such as Decision Tree (DT) and Random Forest (RF), across performance measures. The findings indicate that the proposed hybrid model outperformed traditional models such as RF, which reached an accuracy of 93.5%. This validates the efficacy of the hybrid technique in precisely identifying diverse types of intrusions within IoT systems.

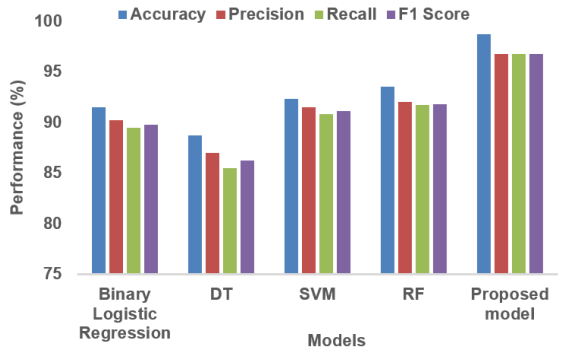


Fig. 9. Comparison of traditional vs. the proposed model.

IV. CONCLUSIONS

This study presented a hybrid DL-based IDS that combines CNN, TCN, CapsuleNet, and LSTM to enhance detection accuracy for all classes in the NSL-KDD dataset. The proposed model achieved enhanced performance by integrating spatial and temporal feature extraction with hierarchical representation and sequential learning, particularly in identifying minority classes, such as R2L and U2R. Utilizing SMOTE for data augmentation substantially reduced overfitting and enhanced generalization. The system achieved a maximum accuracy of 98.71% and maintained high precision, recall, and F1 scores across all categories. In the future, the model can be evaluated using real-time IoT network traffic or other benchmark datasets. Integrating federated learning or edge-computing frameworks may facilitate deployment in resource-limited settings. Furthermore, incorporating explainable AI methods may enhance confidence and transparency in practical cybersecurity implementations, aiding human analysts in understanding and validating intrusion detection decisions.

REFERENCES

[1] M. Benmalek and A. Seddiki, "Particle swarm optimization-enhanced machine learning and deep learning techniques for Internet of Things intrusion detection," *Data Science and Management*, Feb. 2025, https://doi.org/10.1016/j.dsm.2025.02.005.

[2] S. Jamshidi, A. Nikanjam, K. W. Nafi, F. Khomh, and R. Rasta, "Application of deep reinforcement learning for intrusion detection in Internet of Things: A systematic review," *Internet of Things*, vol. 31, May 2025, Art. no. 101531, https://doi.org/10.1016/j.iot.2025.101531.

[3] A. K. Silivery, K. R. M. Rao, and R. Solleti, "Dual-path feature extraction based hybrid intrusion detection in IoT networks," *Computers and Electrical Engineering*, vol. 122, Mar. 2025, Art. no. 109949, https://doi.org/10.1016/j.compeleceng.2024.109949.

[4] T. Devapriya, V. Ganesan, and S. Velmurugan, "Efficient malicious node detection in wireless sensor networks using Rabin-Karp algorithm," *International Journal of Advances in Signal and Image Sciences*, vol. 10, no. 2, pp. 24–36, Dec. 2024, https://doi.org/10.29284/ijasis.10.2.2024.24-36.

[5] M. A. Talukder, M. Khalid, and N. Sultana, "A hybrid machine learning model for intrusion detection in wireless sensor networks leveraging data balancing and dimensionality reduction," *Scientific Reports*, vol. 15, no. 1, Feb. 2025, Art. no. 4617, https://doi.org/10.1038/s41598-025-87028-1.

[6] H. Chen, Z. Wang, S. Yang, X. Luo, D. He, and S. Chan, "Intrusion detection using synaptic intelligent convolutional neural networks for dynamic Internet of Things environments," *Alexandria Engineering Journal*, vol. 111, pp. 78–91, Jan. 2025, https://doi.org/10.1016/j.aej.2024.10.014.

[7] R. Kumar and M. Swarnkar, "QuIDS: A Quantum Support Vector machine-based Intrusion Detection System for IoT networks," *Journal of Network and Computer Applications*, vol. 234, Feb. 2025, Art. no. 104072, https://doi.org/10.1016/j.jnca.2024.104072.

[8] S. P. Maniraj, C. S. Ranganathan, and S. Sekar, "Securing web applications with owasp zap for comprehensive security testing," *International Journal of Advances in Signal and Image Sciences*, vol. 10, no. 2, pp. 12–23, Dec. 2024, https://doi.org/10.29284/ijasis.10.2.2024.12-23.

[9] H. Zeghida *et al.*, "Enhancing IoT cyber attacks intrusion detection through GAN-based data augmentation and hybrid deep learning models for MQTT network protocol cyber attacks," *Cluster Computing*, vol. 28, no. 1, Nov. 2024, Art. no. 58, https://doi.org/10.1007/s10586-024-04752-5.

[10] R. K. Vanakamamidi, L. Ramalingam, N. Abirami, S. Priyanka, C. S. Kumar, and S. Murugan, "IoT Security Based on Machine Learning," in *2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, Aug. 2023, pp. 683–687, https://doi.org/10.1109/SmartTechCon57526.2023.10391727.

[11] A. Almotairi, S. Atawneh, O. A. Khashan, and N. M. Khafajah, "Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models," *Systems Science & Control Engineering*, vol. 12, no. 1, Dec. 2024, Art. no. 2321381, https://doi.org/10.1080/21642583.2024.2321381.

[12] N. U. Bhanu, S. R. Mallick, S. R. Chappidi, and K. Sangeethalakshmi, "RF-SFAD: A Random Forest Model for Selective Forwarding Attack Detection In Mobile Wireless Sensor Networks," *International Journal of Advances in Signal and Image Sciences*, vol. 11, no. 1, pp. 104–116, Jun. 2025, https://doi.org/10.29284/ijasis.11.1.2025.104-116.

[13] B. R. Kikissagbe and M. Adda, "Machine Learning-Based Intrusion Detection Methods in IoT Systems: A Comprehensive Review," *Electronics*, vol. 13, no. 18, Jan. 2024, Art. no. 3601, https://doi.org/10.3390/electronics13183601.

[14] M. Amru *et al.*, "Network intrusion detection system by applying ensemble model for smart home," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 3, pp. 3485-3494, Jun. 2024, https://doi.org/10.11591/ijece.v14i3.pp3485-3494.

[15] A. H. A. Saq, A. Zainal, B. A. S. Al-Rimy, A. Alyami, and H. A. Abosaq, "Intrusion Detection in IoT using Gaussian Fuzzy Mutual Information-based Feature Selection," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17564–17571, Dec. 2024, https://doi.org/10.48084/etasr.8268.

[16] S. Sekar *et al.*, "Intrusion detection and prevention using Bayesian decision with fuzzy logic system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 15, no. 1, pp. 1200–1208, Feb. 2025, https://doi.org/10.11591/ijece.v15i1.pp1200-1208.

[17] A. A. A. Mohammed, "Improving Intrusion Detecction Systems by using Deep Learning Methods on Time Series Data," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19267–19272, Feb. 2025, https://doi.org/10.48084/etasr.9417.

[18] K. Alemerien, S. Al-suhemat, and M. Almahadin, "Towards optimized machine-learning-driven intrusion detection for Internet of Things applications," *International Journal of Information Technology*, vol. 16, no. 8, pp. 4981–4994, Dec. 2024, https://doi.org/10.1007/s41870-024-01852-8.

[19] A. Alsajri and A. Steiti, "Intrusion Detection System Based on Machine Learning Algorithms: (SVM and Genetic Algorithm)," *Babylonian Journal of Machine Learning*, vol. 2024, pp. 15–29, Jan. 2024, https://doi.org/10.58496/BJML/2024/002.

[20] "NSL-KDD." Kaggle, [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd.

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, https://doi.org/10.1613/jair.953.

[22] M. Jain and A. Srihari, "Comparison of Machine Learning Algorithm in Intrusion Detection Systems: A Review Using Binary Logistic Regression." Authorea, May 27, 2025, https://doi.org/10.22541/au.174837862.20090642/v1.