International Journal of Safety and Security Engineering
Vol. 16, No. 2, February, 2026, pp. 249-262
Journal homepage: http://iieta.org/journals/ijsse

Proximity-Based Trust Classification for Private and Public IoE Devices Using a Hybrid CNN–LSTM Architecture

Jayashree C. Pasalkar [1,2*], Dattatraya S. Bormane[3]

[1] Department of Computer Engineering, AISSMS College of Engineering, Savitribai Phule Pune University, Pune 411001, India
[2] Department of Information Technology, AISSMS Institute of Information Technology, Savitribai Phule Pune University, Pune 411001, India
[3] Department of Electronics and Telecommunications, AISSMS College of Engineering, Savitribai Phule Pune University, Pune 411001, India

Corresponding Author Email: jayashree.pasalkar@aissmsioit.org

https://doi.org/10.18280/ijsse.160201

Received: 10 December 2025
Revised: 2 February 2026
Accepted: 15 February 2026
Available online: 28 February 2026

Keywords:
Internet of Everything, trust classification, proximity-based trust, hybrid CNN–LSTM, Social Internet of Things, Local Interpretable Model-agnostic Explanations

ABSTRACT

Trust assessment is essential for enabling reliable collaboration among heterogeneous devices in Internet of Everything (IoE) environments, where dynamic interactions and mobility patterns make trust estimation difficult. This study proposes a trust-classification framework that combines proximity-based trust computation with a lightweight hybrid CNN–LSTM architecture, termed HyLite, for classifying private and public IoE devices. Experiments were conducted on a Social Internet of Things (SIoT) dataset containing 16,216 devices, including 14,600 private devices and 1,616 public devices. The proposed framework first derives trust-related labels from proximity, interaction duration, device-type similarity, and prior trust history, and then learns discriminative spatial-temporal patterns through convolutional and recurrent layers. Comparative evaluation against conventional machine learning and baseline deep learning models shows that HyLite achieved an average k-fold accuracy of 99.7% with a macro F1-score of 99.7% on private devices, and 99.89% accuracy with a macro F1-score of 91% on public devices. To improve interpretability, Local Interpretable Model-agnostic Explanations (LIME) were used to identify the features contributing most strongly to trust decisions. The results indicate that combining proximity-aware trust computation with hybrid deep learning can provide effective trust classification in heterogeneous IoE settings. The study also highlights the need for further validation on larger and more balanced datasets to assess robustness and generalizability.

1. INTRODUCTION

The Internet of Everything (IoE) has an imaginary vision of a pervasive network where personal and governmental objects can communicate and collaborate to provide services within areas like smart cities, healthcare, transportation, and industry. Trust between heterogeneous IoE nodes is essential for ensuring reliable communication and cooperation. In the absence of proper trust assessment, rogue devices are able to interfere with services, steal data, or use network resources. The current security measures can counter particular cyberattacks like denial of service, spoofing, or malware, but trust management needs to evaluate the behavioral and social aspects that go beyond the traditional security measures. A trust management system must dynamically compute trust values, handle large volumes of data, adapt to behavioral changes, and maintain computational efficiency [1, 2].

1.1 Social Internet of Things dataset

To assess trust models accurately, researchers require representative datasets that reflect device diversity and user behavior. The Social Internet of Things (SIoT) dataset employed in this work includes 16,216 objects: 14,600 private devices owned by individuals and 1,616 public devices managed by municipal services. Each object record specifies the device identifier, user identifier, device type, brand, and model. Private devices: smartphones, cars, tablets, fitness wearables, smart watches, personal computers, printers, and home sensors. Ownership percentages show that 91% of all users of smartphones own one, 55% own cars, and 84% own personal computers. The dataset is simulated in the Small World in Motion (SWIM) model, which has 4,000 users, a radius of perception of 0.015, a simulation time of ten days, and a mobility parameter of 0.9. Public facilities comprise points of suggestion, environmental sensors, transport vehicles, digital signage, garbage trucks, street lights, parking detectors, and alarm systems. The data set characterizes fixed and mobile devices by position and time-stamped mobility trajectories; the relationships between people, e.g., ownership, co-location, and device type, are described by the adjacency

matrices [3].

1.2 Trust management principles

Trust management assesses the credibility of the IoE participants through the analysis of the measurable characteristics (e.g., service quality, availability) and social parameters. The trust value can be calculated as a weighted sum of Quality of Service (QoS) trust and social trust, where QoS trust pertains to competence, reliability, and job execution, while social trust relates to the relationships among device owners. Trust scores have to be constantly updated in a dynamic IoE environment, depending on the recent interactions. The combination of various trust attributes has been carried out through multi criteria decision making techniques. Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks are among the Deep Learning (DL) models that have been used to detect device behavior changes and determine trustworthiness. According to recent studies, the accuracy is over 99% when applying LSTM based models on trust management [4, 5].

1.3 Motivation and related work

Traditional systems of trust management often rely on either statistical methods or simple machine learning classifiers. These approaches are effective with small data sets, but they face challenges when there is large dimensionality and temporal variation of the IoE data. Both long-term dependencies and local patterns can be captured by advance / hybrid architectures. Also, many existing models treat all devices uniformly and do not consider differences between private and public devices. Public devices tend to be on the spot and belong to the municipality, whereas private devices are mobile and user-related. These dissimilarities have an impact on the way trust ought to be computed and refreshed. Besides, explainability is a topic that is seldom discussed: deep models are black boxes, so device owners and administrators are not sure what it is that makes features produce trust decisions [6, 7].

1.4 Contributions

Based on this, the paper will suggest a detailed trust classification framework that will be specific to heterogeneous IoE networks. These are the main contributions, which can be summed up as follows:

1.4.1 Proximity based trust calculation

A trust value is computed about each device, taking into consideration its proximity to other devices, co-location relationship, device type, and mobility patterns. There are trust values (trusted, neutral, and untrusted) that are grouped together in supervised learning.

1.4.2 HyLite model

The HyLite model is a hybrid model (CNN – LSTM) with a lightweight architecture that extracts local patterns using convolutional layers and temporal dependencies using LSTM layers. HyLite stands for Hybrid Lightweight, representing a hybrid deep learning architecture designed with a lightweight structure to enable efficient and accurate deployment in resource-constrained IoE environments. This model has fewer filters and dropout regularization, which helps to make the model simpler and less overfitting. To trust the decision made by the explainable Artificial Intelligence (AI), the explainable AI method known as Local Interpretable Model-agnostic Explanations (LIME) is used to interpret the contribution of individual features.

1.4.3 Evaluation on Social Internet of Things dataset

Wide experiments are carried out on the subset of private and public devices of the SIoT dataset. The HyLite model is juxtaposed with the naïve Bayes classifier. The HyLite model demonstrates superior accuracy, precision, and recall, accompanied by reduced loss values.

1.4.4 Analysis of trust dynamics

The model examines the impact of mobility and the type of device on the trust scores. The analysis indicates the stability of the trust score of public devices as they are predictable in their actions, whereas the score of the private devices is more unpredictable.

1.5 Proposed objectives

- Develop an optimized trust classification framework that combines proximity-based trust computation with a HyLite model and LIME based interpretability.
- Validate the proposed framework by comparing it with baseline models on the SIoT dataset and reporting performance metrics for private and public devices.

The subsequent sections of the paper are structured as follows. Section 2 reviews the existing literature, while Section 3 elaborates on the methodology, encompassing trust computations, model design, evaluation procedures, and pseudocode for the proposed algorithm. Using visual aids, Section 4 discusses the experimental outcomes. Section 5 presents the findings and recommendations for additional research.

2. LITERATURE REVIEW

The recent studies on trust management and security in IoT/IoE are summarized in Table 1. The papers learn intrusion detection based on machine learning, trust scoring, and cryptographic protocols solutions. The list of key contributions, methodologies, and performance metrics is provided, and the gaps in research are observed.

Recent studies on IoT security and trust management increasingly employ deep learning models to capture complex behavioral patterns in network data. For example, Khatoon et al. [8] and Alghofaili and Rassam [9] integrate multi-criteria decision making with LSTM networks to evaluate trust relationships among IoT entities, reporting high detection accuracy using packet-level trust features. Similarly, Aaqib et al. [10] and Anwar et al. [11] utilize hybrid deep learning models combining CNN and LSTM to improve trust prediction by capturing both spatial and temporal dependencies in IoT data. While these approaches demonstrate promising performance, their reliance on sequential deep architectures increases computational complexity and limits transparency in how individual features contribute to the final decision. Comparable challenges are also observed in intrusion detection frameworks such as the CNN–LSTM model proposed by Sinha et al. [12] and the CNN–LSTM–GRU architecture introduced by Kilichev et al. [13], where

multiple deep layers enhance prediction capability but often result in heavier models that are difficult to interpret and deploy efficiently in large-scale or resource-constrained IoT environments.

Other works focus on strengthening security through complementary mechanisms such as cryptographic frameworks and trust-based management systems. Pawar et al. [14] and Karunkuzhali et al. [15] emphasize lightweight cryptographic and attribute-based encryption approaches to improve confidentiality and data integrity in IoT and healthcare networks, while Ullah et al. [16] propose an adaptive trust and reputation framework to identify malicious devices in dynamic IoT environments. In addition, Eshmawi et al. [17] employ ConvLSTM for intrusion detection in industrial IoT settings, demonstrating the effectiveness of temporal feature extraction for network traffic analysis. Meanwhile, Ghuraybi et al. [18] integrate blockchain, physical unclonable functions, and machine learning to enhance authentication in cyber-physical systems. Although these studies contribute important advances in security, authentication, and trust management, they often address these aspects in isolation and provide limited discussion on the interpretability of learned features or the broader analytical insights derived from IoT data.

Table 1. Comparative analysis of recent IoT/IoE trust and security research

Author	Methodology	Domain	Key Findings	Dataset / Environment
Khatoon et al. [8]	Machine-learning-based detection and prevention using multi-criteria decision making and LSTM	Cybersecurity for IoE	Proposed a trust management model employing SMART and LSTM; achieved 99.87% accuracy and 99.76% F-measure	Real IoT packet captures with multi-attribute trust features
Alghofaili and Rassam [9]	Multi-criteria decision making combined with LSTM for trust management	IoT services	Proposed SMART-LSTM model achieving 99.87% accuracy and 99.76% F-measure	Extracted dataset with IoT packet captures
Aaqib et al. [10]	Hybrid deep learning model integrating CNN and LSTM for IoT trust management	IoT trust management	Reported improved accuracy over standalone LSTM; details unavailable due to restricted access	Custom IoT dataset with trust labels
Anwar et al. [11]	LSTM integrated with multi-criteria decision-making (IoTGuard framework)	IoT trust management	Employed LSTM and multi-criteria decision making; reported high detection accuracy (exact numbers unavailable)	IoTGuard environment
Sinha et al. [12]	Secure hybrid architecture combining LSTM and CNN for IoT scenarios utilising deep learning	IoT intrusion detection	Achieved 99.87% accuracy, 99.89% precision and 99.85% recall; maintained 90.2% accuracy under adversarial conditions	BoT-IoT dataset
Kilichev et al. [13]	CNN–LSTM–GRU integrated model for electric vehicle charging stations	IoT security	Developed an intrusion detection model integrating CNN, LSTM and GRU; improved accuracy compared to individual models	EV charging station dataset
Pawar et al. [14]	Blockchain-facilitated cybersecurity utilising elliptic curve cryptography and the black-winged kite model	IoT security	Introduced a lightweight cryptographic approach to ensure confidentiality and integrity; emphasised reduced overhead	Simulated IoT network
Karunkuzhali et al. [15]	A combination of attribute-based encryption and hybrid lightweight cryptography for wireless body area networks	Healthcare IoT	Developed a secure health monitoring system using attribute-based encryption; improved confidentiality and reduced power consumption	Wireless body area network
Ullah et al. [16]	Comprehensive trust framework for adaptive trust and reputation management	IoT networks	Proposed dynamic trust and reputation management; demonstrated improved detection of malicious devices	Simulated IoT network
Eshmawi et al. [17]	ConvLSTM-based intrusion detection in industrial IoT and cloud computing	Industrial IoT	Developed a ConvLSTM model for network intrusion detection; reported high detection rate	Industrial IoT dataset
Ghuraybi et al. [18]	Integration of blockchain technology, physical unclonable functionalities, and machine learning for authentication purposes	Cyber-physical systems	Proposed multi-layered authentication scheme combining blockchain and machine learning; improved resistance against spoofing attacks	Cyber-physical system prototype

Note: Internet of Things (IoT), Internet of Everything (IoE), Social Internet of Things (SIoT), Machine Learning (ML), Deep Learning (DL), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM).

This indicates the continuing need for approaches that not only achieve strong predictive performance but also offer a clearer understanding of feature influence and decision-making behavior within IoT security models. According to research, deep learning models are capable of achieving very accurate results in trust management and intrusion detection. However, several gaps remain:

Lack of differentiation between device types: Most of the research uses all types of IoT devices in a homogenous manner without regard to behavioral differences between private and public devices. In this paper, these categories are differentiated and it has been shown that individual models perform better.

• Limited explainability: Deep models that are present are black boxes. There are not many works that combine explainable AI tools like LIME to understand trust decisions. Our framework deals with this through the feature-level explanation.

• Proximity and social factors: While some studies use

multi-criteria decision making, few incorporate proximity relationships and social interactions as trust features. This work calculates trust based on co-location frequency, interaction duration, and device type similarity.

• Realistic dataset evaluation: Many models are evaluated on synthetic datasets or small test beds. The SIoT dataset with over 16,000 devices and real mobility traces provides a more realistic benchmark. The proposed framework achieves high accuracy on this challenging dataset.

The suggested trust classification framework fills these gaps, which in turn aids in the creation of trustworthy IoE systems and lays the groundwork for studies of adaptive and explicable trust management in the future.

3. METHODOLOGY

3.1 Dataset description

The dataset is retrieved from the Net4U SIoT repository. Records are separated into private and public device subsets based on ownership [19]. Missing values are filled by forward filling for time series data and by discarding incomplete records, as shown in Figures 1 and 2. Each device's trajectory is represented as a sequence of time-stamped events containing device type, location coordinates, interactions, and service requests. The sequences are segmented into fixed-length windows for feature extraction.

	timestamp_start	timestamp_stop	id_user	x	y
0	1728020	1728950	685	0.741629	0.581316
1	1728030	1728270	1805	0.253081	0.997948
2	1728040	1728190	1953	0.909359	0.378448
3	1728040	1728560	2449	0.945438	0.629264
4	1728040	1728890	2332	0.296147	0.222340

Figure 1. Mobile devices dataset sample (private devices)

	timestamp_start	timestamp_stop	id_device	x	y
0	13.424611	273.046328	16182	0.579087	0.620131
1	25.657326	218.264436	16180	0.548562	0.510441
2	43.745912	303.809689	16171	0.494062	0.466788
3	52.129195	339.227927	16207	0.592766	0.527319
4	52.766923	281.245891	16177	0.484612	0.334934

Figure 2. Mobile devices dataset sample (public devices)

3.2 Model architecture based trust calculation

The methodology comprises data preparation, trust calculation, formation of trust groups, model architecture, training, and evaluation. Detail System Architecture diagram is shown in Figure 3.

3.3 Proximity based trust calculation

Let $D = \{d1, d2, \dots, dN\}$ denote the set of devices and $P(d_i)$ represent the set of neighboring devices within a proximity radius r. For device d_i, the trust value $T(d_i)$ is computed using weighted contributions of co-location frequency, interaction duration, device type similarity and prior trust history as shown in Eq. (1):

$$(T(d_i) = \frac{1}{|P(d_i)|}\sum_{d_j \in P(d_i)}[w_1 . f_{coloc}(d_i, d_j) + w_2 . f_{dur}(d_i, d_j) + w3 . f_{type}(d_i, d_j) + w4 . T_{prev}(d_j)] \quad (1)$$

where, f_{coloc} "counts the number of co-locations", f_{dur} " measures average interaction duration", f_{type} "equals 1 if devices share the same type and 0 otherwise", $T_{prev}(d_j)$ denotes "the previous trust score of neighbor d_j ", and $w1, w2, w3, w4$ are weights satisfying $w1 + w2 + w3 + w4 = 1$ [20]. The proximity radius r is selected according to the communication ranges of the technologies specified in the dataset (Bluetooth: 40 m, Wi-Fi: 400 m and LoRa: 1500 m).

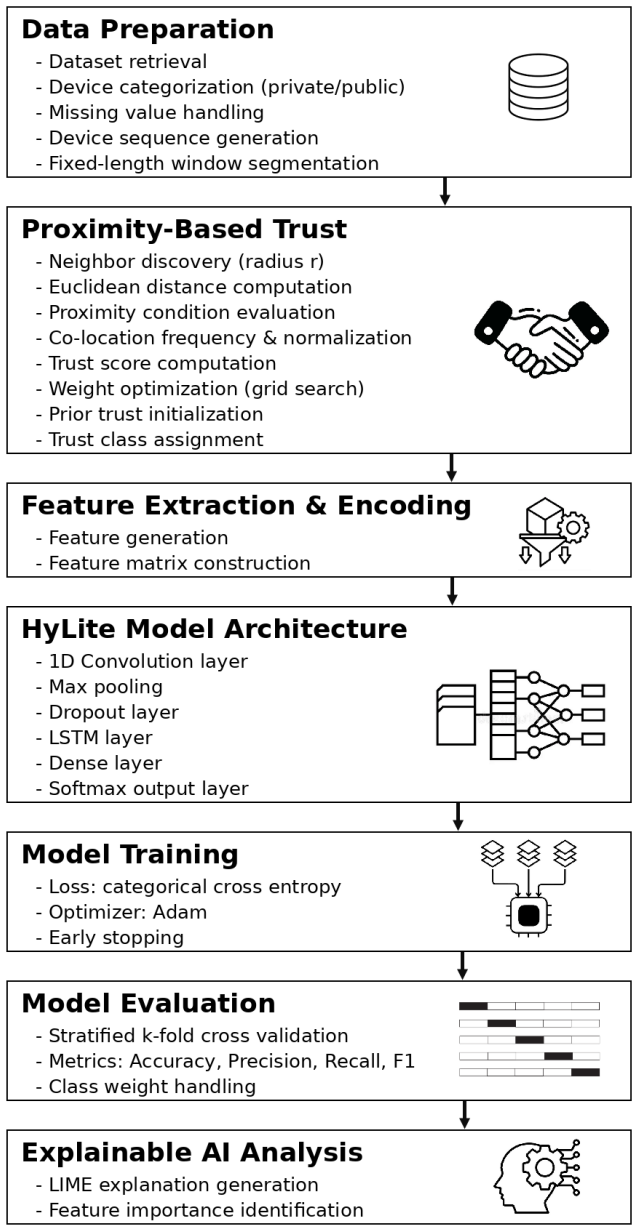


Figure 3. Proposed system architecture
Note: One Dimension (1D), Long Short-Term Memory (LSTM), Local Interpretable Model-agnostic Explanations (LIME).

Computation of f_{coloc}

From raw location data, the SIoT dataset contains timestamped coordinates for each device. Co-location was

derived using spatial proximity and time overlap:

Step 1. Euclidean distance between two devices at time *t*

$$d_{ij}(t) = \sqrt{[\left(x_i(t) - x_j(t)\right)^2 + \left(y_i(t) - y_j(t)\right)^2]} \quad (2)$$

Step 2. Proximity condition for co-location

$$C_{ij}(t) = \begin{cases} 1, & if\ d_{ij}(t) \le r \\ 0 & Otherwise \end{cases} \quad (3)$$

Step 3. Co-location frequency between devices d_i and d_j

$$f_{coloc}(d_i, d_j) = \sum_{t=1}^{T} C_{ij}(t) \quad (4)$$

Step 4. Normalized co-location score

$$f_{coloc}(d_i, d_j) = \frac{1}{T}\sum_{t=1}^{T} C_{ij}(t) \quad (5)$$

where,

- $x_i(t), y_i(t)$ epresent the spatial coordinates of device d_i at time *t*,
- *r* denotes the communication proximity radius,
- *T* represents the total number of observed timestamps, and
- $C_{ij}(t)$ indicates whether devices d_i and d_j are co-located at time t.
- f_{coloc} represents Co-location frequency.

Weight selection (*w*1 to *w*4)

The weights are determined through grid-search optimization on the training portion of the SIoT dataset. Candidate combinations were generated under the constraint $w1 + w2 + w3 + w4 = 1$ with a step size of 0.1. Each combination was evaluated using cross-validation based on the classification accuracy of the downstream CNN–LSTM (HyLite) model.

The best performing configuration was:

*w*1 = 0.40 (co-location frequency)
*w*2 = 0.30 (interaction duration)
*w*3 = 0.15 (device type similarity)
*w*4 = 0.15 (prior trust score)

Higher weight was assigned to co-location and interaction duration because mobility interactions in the SIoT dataset strongly correlate with trust behaviour. This selection process ensures the weights are empirically derived rather than manually guessed.

Prior trust score initialization

The initial trust score $T_{prev}(d_j)$ is set to 0.5 for all devices at the first iteration and updated iteratively using previously computed trust values.

To categories devices into trust groups, thresholds θ_1 $and\ \theta_2$ are applied: devices with $T(d_i) \ge \theta_2$ are considered trusted, those with $\theta_1 < T(di) < \theta_2$ are neutral, and those with $T(di) \le \theta_1$ are untrusted. The thresholds are established empirically according to the distribution of trust scores within the training set.

3.4 Feature extraction and model architecture

Once trust labels are assigned, each device sequence is encoded into a two-dimensional matrix, where rows represent time steps and columns represent features such as location coordinates, interaction counts, device type indices, and computed trust values.

The architecture proposed in Figure 4 is a combination of convolutional, recurrent, and dense layers to determine the reliability of IoE devices. The first is an input layer that provides the time-series features of proximity, interaction time, and device type. In the first stage, the one-dimensional convolution is performed using 16 filters and a three-sized kernel. This procedure gathers long-range patterns from adjacent time steps, subsequently employing max-pooling to diminish dimensionality and a dropout layer to alleviate overfitting.

The pooled features are then convolved again with 32 filters, and once more, the process is repeated with pooling and dropout. Once local patterns are extracted, the sequence goes to LSTM that is used to model dependencies in the long-term and the temporal context. The LSTM output passes through a fully connected layer of 32 units, which refines the feature representation and adds another dropout. The last thick layer is made with the help of the softmax activation and generates the probabilities of the three trust classes: trusted, neutral, and untrusted. It is a series of layers that will enable the model to identify both short-term and developing behavioral trends. Convolutional layers act as feature extractors, LSTM layers are used for temporal dynamics, while dense layers map learned patterns into class probabilities. The dropout at several stages decreases the chances of overfitting. The architecture, therefore, is subjecting local pattern recognition with temporal reasoning to make effective trust classifications which aligns with prior studies that have demonstrated that HyLite models are able to exploit spatial and temporal features effectively.

HyLite model architecture summary, as shown in Figure 5. The HyLite model architecture comprises the following layers.

3.4.1 Convolutional layer

A one-dimensional convolution with 16 filters and kernel size of 3 processes the input sequences to extract local patterns.

The activation function employed is (ReLU to incorporate non-linearity. The padding is configured to "same" to ensure the output length corresponds with the input length. A max pooling procedure with a window size of 2 down samples the feature maps following convolution.

3.4.2 Dropout layer

A dropout rate of 0.2 is employed to randomly deactivate neurons during training, hence mitigating overfitting.

3.4.3 LSTM layer

A lengthy short-term memory layer with 32 units captures temporal dependencies in the downsampled feature maps. Only the ultimate output is transmitted to the subsequent layer.

3.4.4 Dense layer

A fully connected layer including 32 neurons processes the acquired characteristics, succeeded by a ReLU activation function.

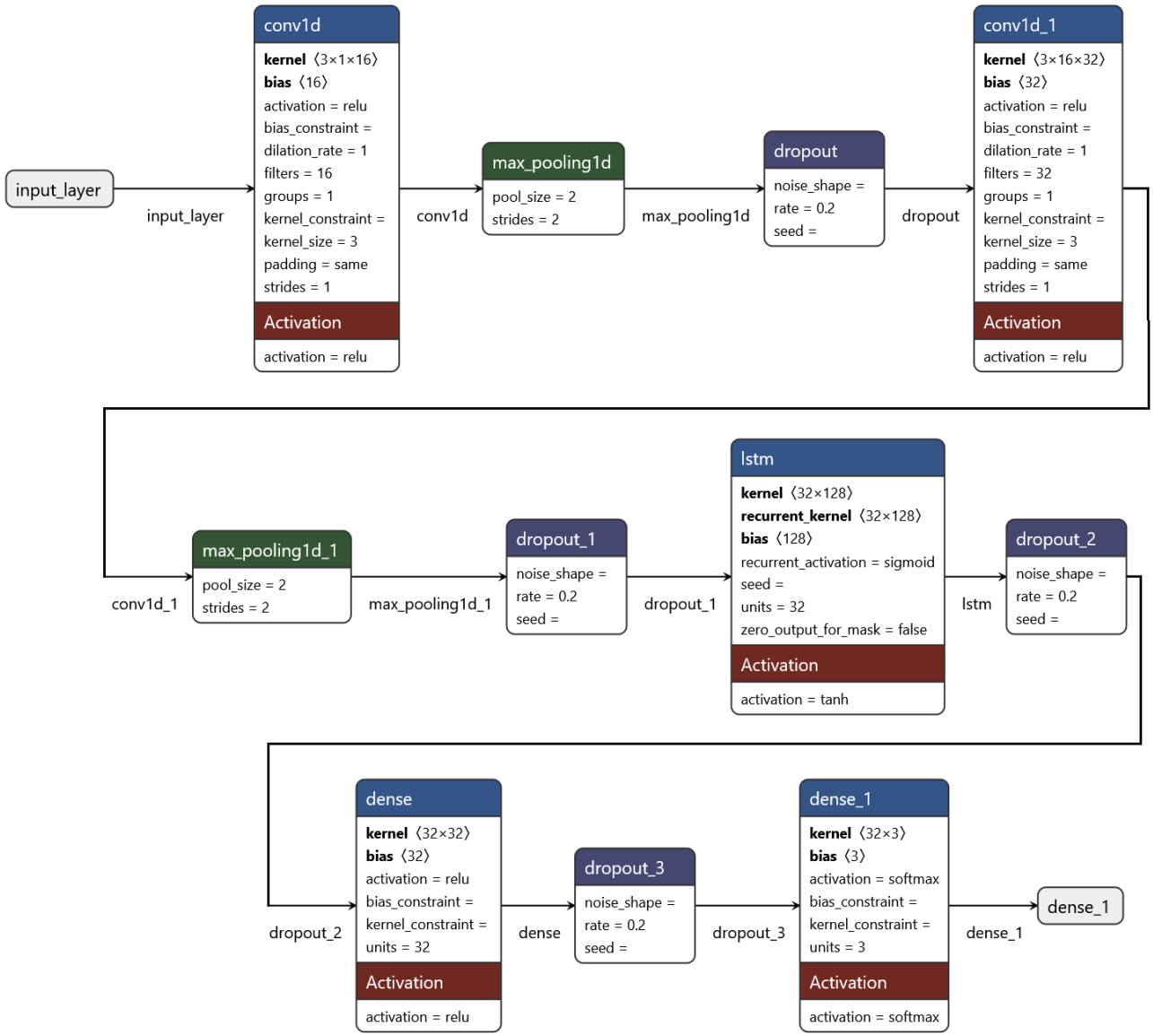


Figure 4. Proposed HyLite model architecture

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 4, 16)	64
max_pooling1d (MaxPooling1D)	(None, 2, 16)	0
dropout (Dropout)	(None, 2, 16)	0
conv1d_1 (Conv1D)	(None, 2, 32)	1,568
max_pooling1d_1 (MaxPooling1D)	(None, 1, 32)	0
dropout_1 (Dropout)	(None, 1, 32)	0
lstm (LSTM)	(None, 32)	8,320
dropout_2 (Dropout)	(None, 32)	0
dense (Dense)	(None, 32)	1,056
dropout_3 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 3)	99

Total params: 11,107 (43.39 KB)
Trainable params: 11,107 (43.39 KB)
Non-trainable params: 0 (0.00 B)

Figure 5. HyLite model architecture summary

3.4.5 Output layer

A softmax layer with three neurons produces trust classification probabilities for the trusted, neutral, and untrusted classes.

The use of convolution followed by LSTM is motivated by the ability of convolutional layers to identify short-term spatial patterns while LSTM layers model long-term temporal relationships. This structure is particularly well-suited to IoE data, which contains repeated local patterns (e.g., periodic user movements) and gradual behavioral changes.

The proposed HyLite model adopts a compact sequential architecture designed for efficient deployment in resource-constrained IoE environments. The model integrates lightweight convolutional layers for feature extraction, pooling and dropout layers for dimensionality reduction and regularization, and an LSTM layer to capture temporal dependencies in the data. The features extracted are passed through fully connected layers to be classified ultimately using a softmax output layer. In general, the HyLite architecture has 11,107 trainable parameters (about 43.39 KB memory footprint), which shows its memory efficiency and performance and its relevance to edge-based IoE applications without any negative impact on predictive performance.

Algorithm 1. Optimized CNN–LSTM Trust Classification

Input: Device sequences X, proximity radius r, weights w1…w4, thresholds θ1, θ2, number of folds k.
Output: Trained CNN–LSTM model, trust labels Y.
1: for each device di ∈ D do
2: Compute neighbor set P(di) within radius r.
3: Calculate trust value T(di) using Equation (1).
4: if $T(di) \geq \theta 2$ then label di as trusted
5: else if $T(di) > \theta 1$ then label di as neutral
6: else label di as untrusted
7: end for
8: Encode sequences and labels to obtain training dataset (X, Y).
9: Initialize CNN–LSTM model parameters Θ.
10: for fold j = 1 to k do
11: Split (X, Y) into training set Dj_{train} and validation set Dj_{val}.
12: Train model on Dj_{train} using cross-entropy loss and Adam optimizer.
13: Evaluate model on Dj_{val} and record metrics $ACCj, PRj, REj, F1j$.
14: end for
15: Average metrics across folds to assess performance.
16: Apply LIME for explainability on selected samples.
17: Return trained model Θ and trust labels Y.

Note: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Local Interpretable Model-agnostic Explanations (LIME).

3.5 Model training

Let $X \in \mathbb{R}^{N \times T \times F}$ denote the training data, where N is the number of sequences, T is the sequence length and F is the number of features. Let Y ∈ $\{0,1,2\}^N$ be the trust labels. The model parameters θ are learned by minimizing the categorical cross-entropy loss represented in Eq. (2):

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=0}^{2} 1[Y_i = c] log p_c(X_i; \theta) \quad (6)$$

where, $pc(X_i; \theta)$ is the anticipated probability of class ccc for sample i. The model is optimised using the Adam algorithm with a learning rate of η = 0.001. Early halting is implemented according to validation loss to avert overfitting.

3.6 Machine learning / deep learning models

K-Nearest Neighbors (KNN) – The KNN [21] algorithm is a supervised machine learning technique employed for classification and regression applications. It is a non-parametric, instance-based approach that predicts the class of a new data point by identifying the majority class among its K nearest neighbours in the training dataset. The resemblance between data points is typically quantified using distance metrics, such as Euclidean distance. When a new sample is introduced, the algorithm calculates the distance between the sample and all training data, selects the closest K neighbors, and assigns the most frequent class among them as the predicted output. KNN is simple to implement and effective for pattern recognition tasks, but its performance depends on the choice of K value and proper feature scaling.

Naïve Bayes (NB): Naive Bayes [21] is a probabilistic machine learning classifier based on Bayes' theorem, which assumes conditional independence among input features. It calculates the posterior probability of each class given the observed features and assigns the instance to the class with the highest probability. Despite its simplicity, Naïve Bayes is highly efficient and performs well in many classification tasks, especially when dealing with high-dimensional datasets. The model requires relatively small training data, is computationally fast, and is widely used in applications such as text classification, spam detection, and medical diagnosis.

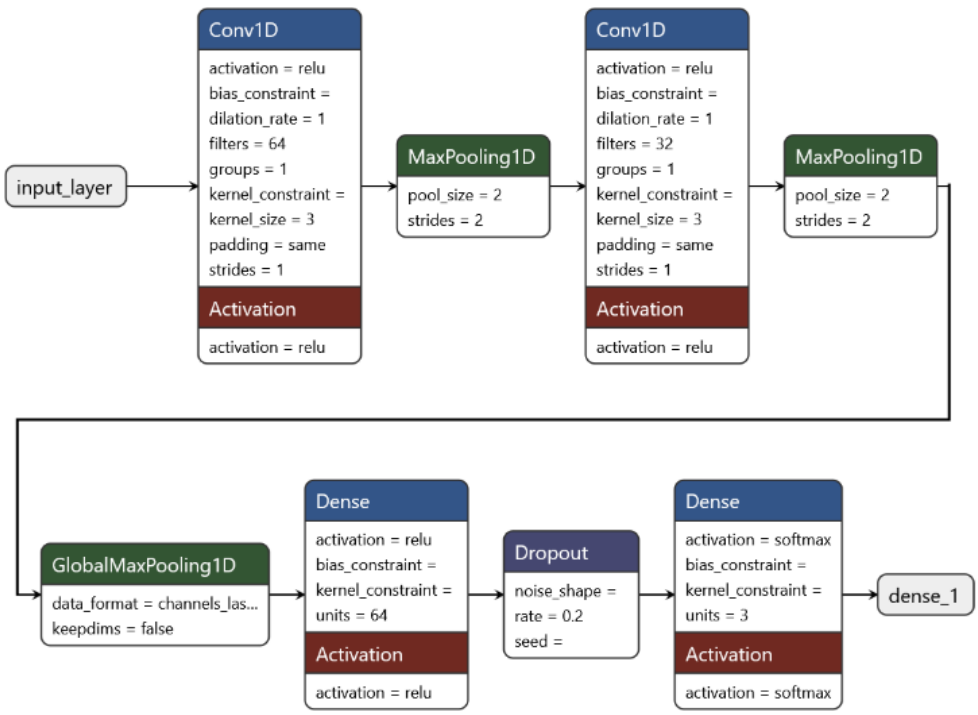


Figure 6. Convolutional Neural Network (CNN) Model architecture

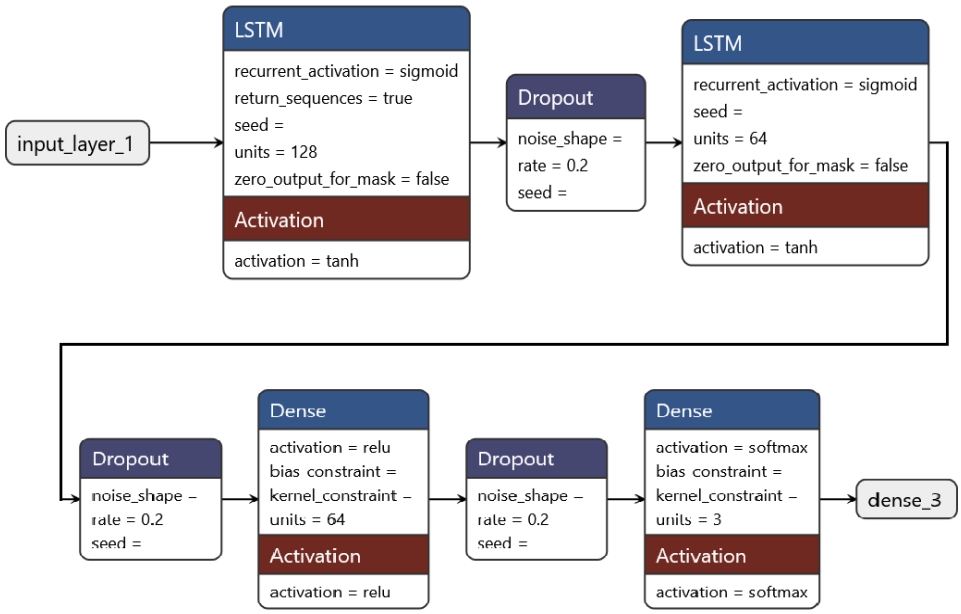


Figure 7. Long Short-Term Memory (LSTM) model architecture

CNN: CNN [22] is a DL model that extracts hierarchical features out of the input data, automatically, using convolution operations. The convolutional layers facilitate the identification of local patterns in CNNs, the pooling layers diminish dimensionality, and the fully connected layers perform the final classification.They are good learning models in the spatial features representations and are widely applied in image analysis, signal processing and sequential data modeling. Figure 6 shows the CNN Model architecture.

LSTM: LSTM [22] is a type of Recurrent Neural Network (RNN) specifically engineered to learn long-term sequential dependencies in data. LSTM networks possess memory cells and gating structures (input gate, forget gate, and output gate) to manage information flow, enabling the model to retain significant temporal patterns while discarding irrelevant data. Time-series prediction, sequential data analysis, and temporal pattern recognition are the areas of application of LSTM that are especially enabled by the mentioned ability. Figure 7 shows the LSTM Model architecture.

3.7 K-fold cross-validation

Stratified k-fold cross-validation is employed to ensure generalizability. The dataset is partitioned into k disjoint folds $\{D_1, \ldots, D_k\}$. For each fold j, the model is trained on D_j and validated on D_j. The average accuracy (ACC), precision (PR), recall (RE) and F1 score across the folds are calculated as per Eqs. (7)-(10):

$$Acc = \frac{1}{k}\sum_{j=1}^{k}\frac{TP_j + TN_j}{TP_j + TN_j + FP_j + FN_j} \tag{7}$$

$$PR = \frac{1}{k}\sum_{j=1}^{k}\frac{TP_j}{TP_j + FP_j} \tag{8}$$

$$RE = \frac{1}{k}\sum_{j=1}^{k}\frac{TP_j}{TP_j + FN_j} \tag{9}$$

$$F1 = \frac{2 \times PR \times RE}{PR + RE} \tag{10}$$

TP, FP, TN, and FN represent "true positives, false positives, true negatives, and false negatives," accordingly. Class imbalance is addressed by using class weights computed from the training data.

3.8 Explainable Artificial Intelligence using Local Interpretable Model-agnostic Explanations

Applying LIME to individual forecasts helps to understand the model's conclusions. By using modified copies of the input to train a basic surrogate model, LIME approximates the complicated model around a given sample. We can use the coefficients of the surrogate model to determine how significant each feature is. Feature importance plots help identify which aspects of proximity, device type or interaction patterns most influence the trust classification [23].

4. RESULT AND DISCUSSION

The performance of the proposed HyLite (Hybrid CNN–LSTM) model was evaluated by comparing it with several baseline machine learning and deep learning models, including Naïve Bayes, KNN, CNN, and LSTM Models, An 80% – 20% train–test split was used for model training and evaluation to ensure reliable performance assessment. In addition, experiments were conducted considering both public and private IoT devices to analyze the effectiveness of the models under different device environments.

4.1 Experimental setup

Table 2 provides an overview of hardware and software implementation and evaluation environment, such as the Google Colab based on the utilization of parallel computers via GPUs, programming language, and major libraries used in the course of the experimentation.

Table 2. Hardware and software requirements

Component	Specification
Development Environment	Google Colab (Cloud-based Jupyter Notebook)
Processor	Intel Xeon CPU (Google Cloud backend)
GPU	NVIDIA Tesla T4 GPU (GPU-enabled Colab runtime)
RAM	~12–16 GB
Operating System	Linux-based Google Cloud environment
Programming Language	Python 3.10
Deep Learning Framework	TensorFlow 2.15
Machine Learning Library	Scikit-learn 1.4
Numerical Computing	NumPy 1.26
Data Processing	Pandas 2.2
Visualization Library	Matplotlib 3.8
Model Development	Keras (TensorFlow backend)
Development Platform	Jupyter Notebook (Google Colab)

Table 3 presents the hyperparameter configuration used for training the proposed HyLite model, including network parameters, optimizer settings, training epochs, batch size, and validation strategy.

Table 3. Hyperparameter configuration

Hyper-parameter	Value / Setting
CNN Layer Filters	16 / 32
CNN Layer Kernel Size	3
Pooling Type	MaxPooling1D
Pooling Size	2
LSTM Units	32
Dense Layer Units	32
Output Activation	Softmax
Hidden Activation Function	ReLU
Dropout Rate	0.2
Optimizer	Adam
Learning Rate	0.001
Loss Function	Categorical Crossentropy
Batch Size	2048
Epochs	10
Early Stopping	Enabled (patience = 3)
Validation Method	Stratified 5-Fold Cross Validation

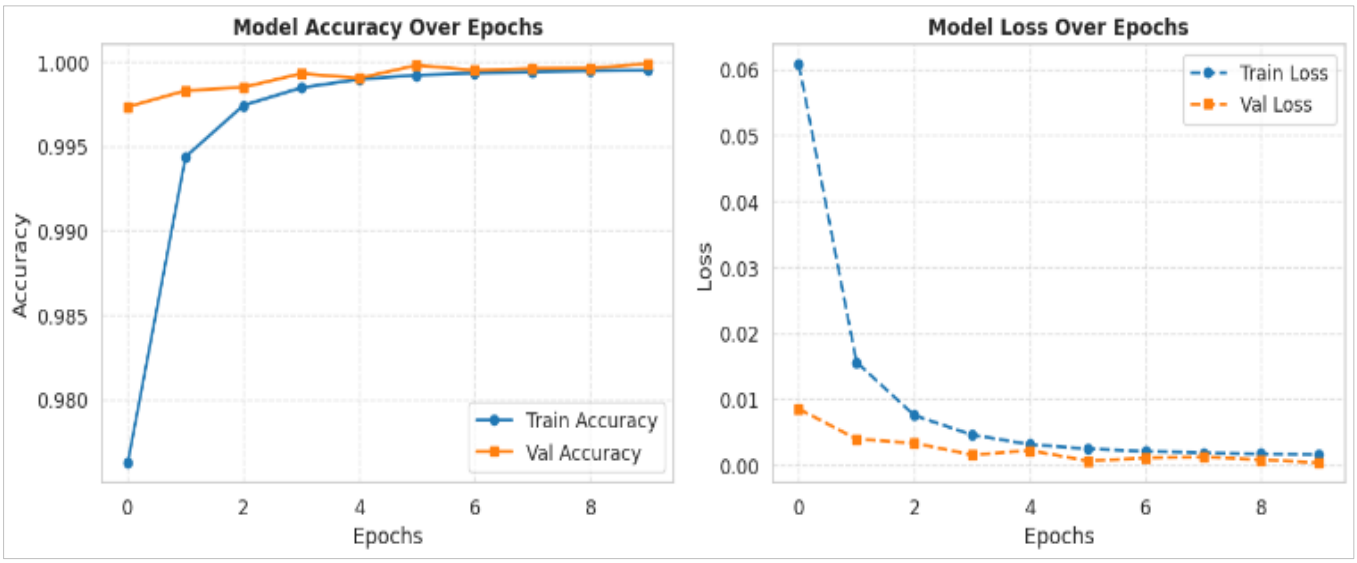
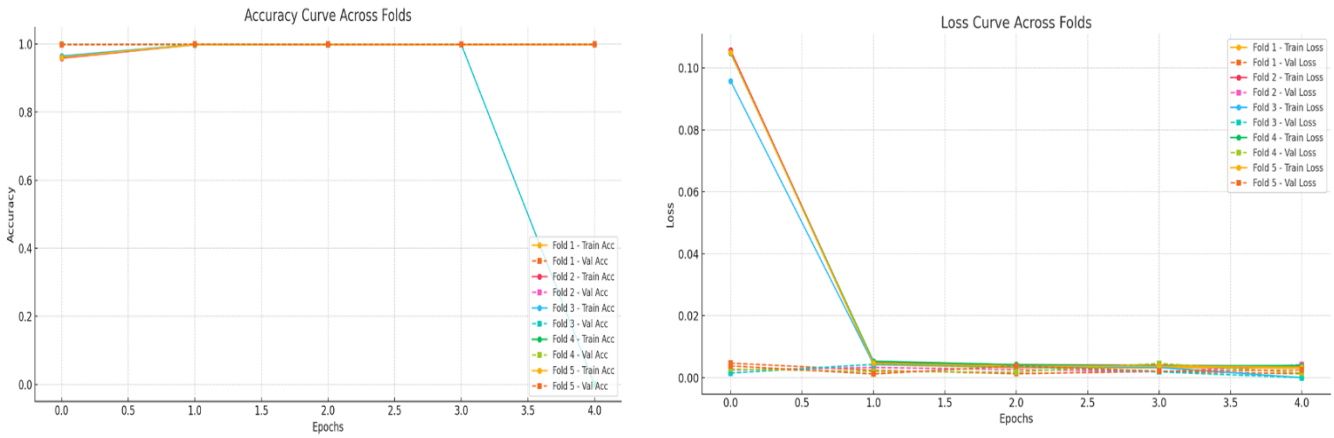


Figure 8. Accuracy and loss curve using HyLite model on private devices



(a) (b)

Figure 9. (a). HyLite model k-fold cross validationvv accuracy graph on private devices; (b). HyLite model k-fold cross validation loss graph on private devices

4.2 Performance on private devices

The HyLite model was trained and evaluated on the subset of 14,600 private devices. Presented in Figure 8 shows the training accuracy and loss curves for a single-fold training system of HyLite Model on private devices. The training accuracy rapidly increased and stabilized around ~99% after 10 epochs, while the loss decreased monotonically, indicating effective learning.

K-fold cross validation accuracy and loss results are shown in Figure 9(a) and 9(b), showing that the HyLite model consistently outperformed machine learning and deep learning model, achieving an average accuracy of 99.7%, macro precision of 99.5%, macro recall of 99.4%, and macro F1-score of 99.7%.

4.3 Explainability analysis for private devices

LIME visualizations were generated for 50 randomly selected private devices. The explanations revealed that proximity features (e.g., number of encounters and interaction duration) contributed most strongly to trust decisions. Device type similarity and prior trust history were less influential. Interestingly, in some misclassified cases, LIME indicated that sudden drops in interaction frequency led the model to assign lower trust scores, reflecting the model's sensitivity to behavior changes.

The LIME visualization for the HyLite model, shown in Figure 10, highlights the key features that influence the model's classification decisions. Unlike simpler probabilistic models, the HyLite architecture captures both feature interactions and sequential patterns, enabling more informative interpretations. In the untrusted class example, the prediction is strongly influenced by the distance range (0.47–0.66) and a large number of inactive devices (e.g., 11,843 inactive neighbors), which significantly contribute toward identifying suspicious network behavior. Additionally, the number of mobile devices (< 844) also affects the prediction, indicating that the model considers device activity patterns when determining trust levels.

Similarly, in the neutral class scenario, the LIME explanation shows that mid-range distance values and a higher number of static devices contribute to the classification, while lower mobile device counts help maintain the neutral prediction. In the trusted class case, the HyLite model demonstrates improved interpretability by balancing multiple features simultaneously, including device counts and distance measures, to produce a confident classification. Overall, the visualization indicates that the HyLite model effectively leverages multiple contextual features to distinguish between trusted, neutral, and untrusted states, demonstrating a more nuanced decision-making process and improved classification capability compared to simpler baseline models.

Figure 10. XAI (Local Interpretable Model-agnostic Explanations (LIME)) result visualization of HyLite model

4.4 Performance on public devices

For the 1,616 public devices, the HyLite model maintained better performance. The training accuracy converged to around ~99.0% after 10 epochs, as shown in Figure 11.

Cross validation results showed an average accuracy of 99.89%, macro precision of 100%, macro recall of 86% and macro F1-score of 91%. The accuracy across all five folds is relatively stable which means that there is consistency in the model behavior in terms of the partitioning of the data. Nonetheless, there are some fluctuations between folds implying sensitivity to feature distribution. While the overall accuracy is considered acceptable, the graph reveals inadequate flexibility regarding the complex interaction patterns among the devices of the population, necessitating a model capable of managing non-linear and temporal dependencies, as illustrated in Figure 11.

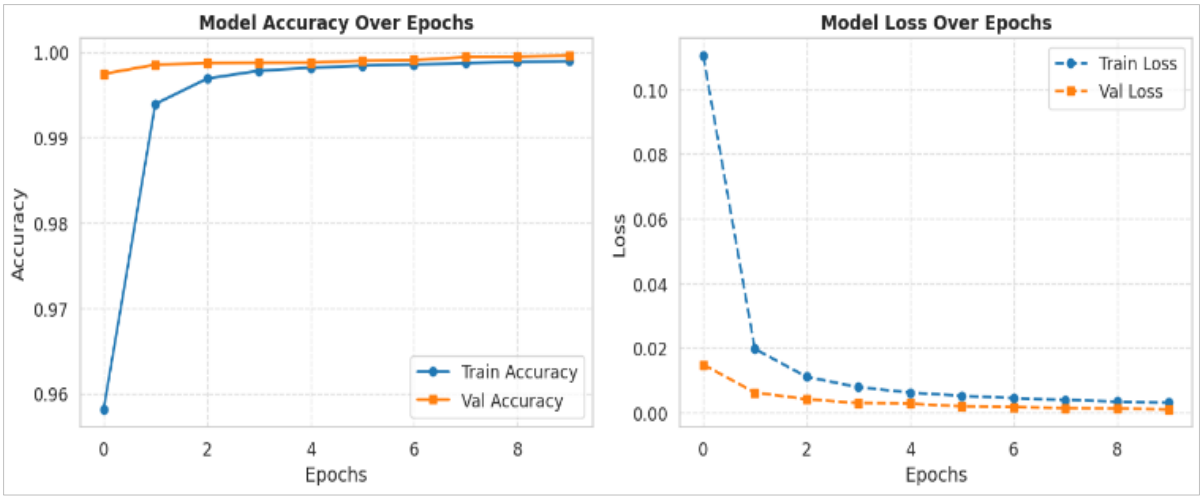
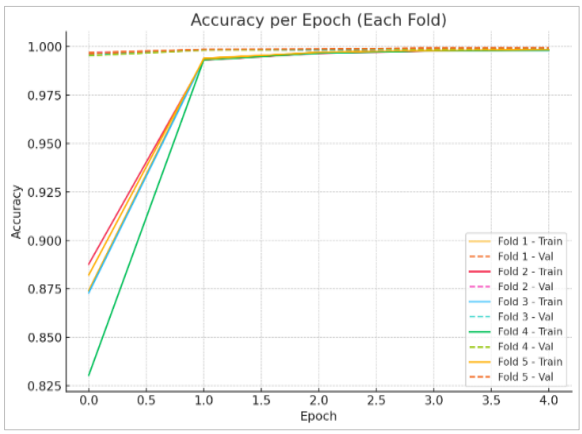


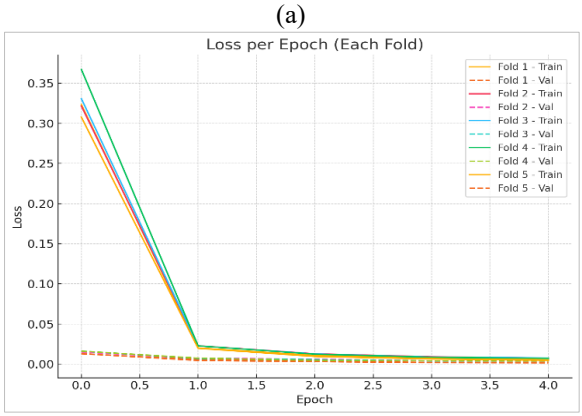
Figure 11. Accuracy and loss curve using the HyLite model on public devices

The graph below illustrates the trend of training accuracy and validation accuracy for the HyLite model throughout 5 folds. The accuracy increases sharply during the initial epochs, after which both the training and validation sets converge to optimal performance levels. Training and validation curves have been close in both folds, which reflects good generalization and a few cases of overfitting. The uniformity among folds leads to the confirmation of the strength of the hybrid system to learn trust pattern-based on public device interactions, which show evident enhancement compared to the traditional ML systems in the same validation conditions as shown in Figure 12(a) and 12(b).

This value illustrates a loss difference between training and validation stages in each of the folds. The loss at the beginning of epochs increases and levels off almost immediately after that at a very small value. The near overlap of the training and validation loss curves indicates effective regularization and stability in convergence. The difference in the folds is minimal indicating that the CNN-LSTM architecture does not change learning stability when there is a change in the different data splits. Figure 11 shows the suggested model in context, and the behavior demonstrates that the planned HyLite model is suitable for identifying the trust problem with heterogeneous public IoE devices.



(a)



(b)

Figure 12. (a). HyLite model k-fold cross validation accuracy graph on public devices; (b). HyLite model k-fold cross validation loss graph on public devices

4.5 Explainability analysis for public devices

The LIME visualization for the HyLite model in Figure 13 illustrates how important features contribute to the classification of public devices. The model assigns the highest probability to the neutral class, while the other classes receive minimal contribution, showing that HyLite identifies this sample as belonging primarily to the neutral trust state. The explanation indicates that the distance value above 0.50 is a major factor driving the prediction, while the counts of static and mobile devices also provide supportive contextual information for the final decision. Overall, this visualization demonstrates that the HyLite model captures meaningful feature contributions in a more structured and reliable manner, allowing clearer interpretation of how trust decisions are formed. Rather than relying on a single simplistic boundary, HyLite uses the combined influence of multiple features to better characterize public-device behavior and produce more robust trust classification.

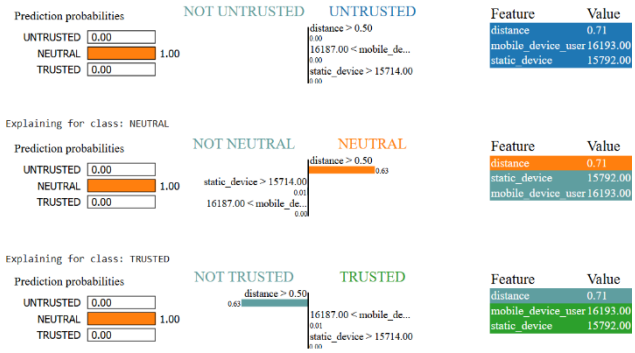


Figure 13. XAI (Local Interpretable Model-agnostic Explanations (LIME)) model result visualization of HyLite model

4.6 Comparative analysis

Table 4 presents the comparative performance of different models on private device data and public device data using evaluation metrics such as accuracy, macro precision, macro recall, and macro F1-score. This comparison helps assess how effectively each method classifies trust levels while maintaining balanced performance across classes.

Table 4. Comparative analysis of models

Model / Method	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
		Private Devices		
KNN	85.4	81.1	76.8	78.8
NB	97.8	98.2	93.9	95.7
CNN	99.0	99.5	96.6	97.9
LSTM	98.0	95.8	98.2	96.9
HyLite	99.7	99.5	99.4	99.7
		Public Devices		
KNN	96.77	97.86	75.68	81.06
NB	96.8	97.9	76.3	81.8
CNN	99.9	99.9	83.6	89.1
LSTM	99.9	99.9	83.2	88.8
Hylite	99.89	100	86	91

Figure 14 and Figure 15 compare the k-fold accuracy values of the Baseline models with HyLite models for both private and public devices. HyLite model was always more accurate in all the folds. The findings are comparable to the current studies that obtained 99.7% accuracy with low false alarms in the IoT intrusion detection using HyLite architectures.

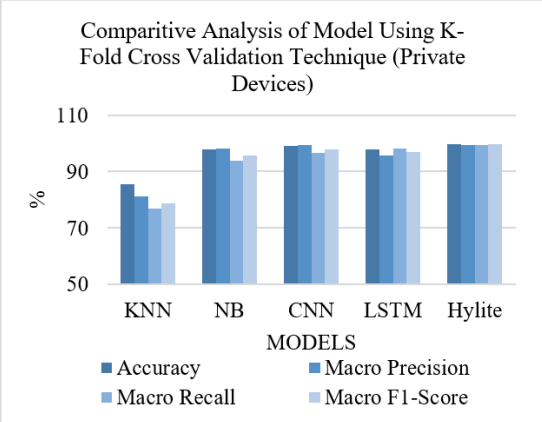


Figure 14. Comparative analysis of the model using k-fold cross validation technique (private devices)

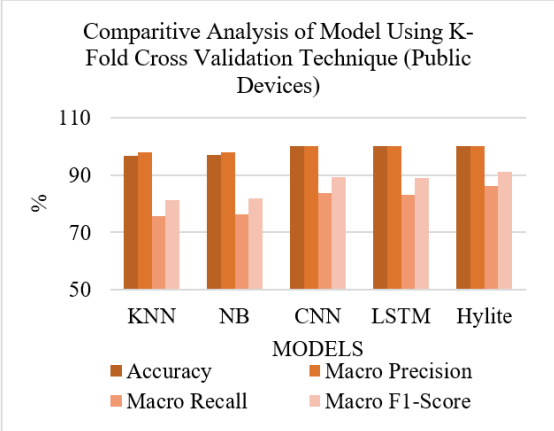


Figure 15. Comparative analysis of the model using the k-fold cross validation technique (public devices)

In the case of a private IoE device, HyLite model is better in performance than baseline models. HyLite (Hybrid CNN-LSTM) attains an accuracy of 99.89%, with macro precision at 100% and, macro recall 86%, and macro F1-score of 91%, thereby validating its efficacy in modeling intricate trust patterns.

4.7 Discussion

4.7.1 Analysis of model performance in public and private Internet of Everything devices

According to the experiment results, the proposed HyLite model performs better on the case of private surveyed IoE devices than on the case of the public. Such a difference can be explained by the characteristics of device behavior in the real-world IoE environment. Personal gadgets generally have fewer uncontrolled and more predictable settings, resulting in more predictable communication patterns and feature of trust, and therefore are simpler to classify as per the model. Conversely, IoE devices used by the general public are exposed to a larger number of users and network conditions, which leads to more heterogeneous and noisy data and can impact classification performance. Besides, the data on this research has some levels of imbalance in classes, which can affect some of its evaluation metrics, especially recall and the F1-score of the minority classes. Nevertheless, HyLite model exhibits high generalization capacity when it comes to the variety of devices. These findings indicate the practical issues in evaluating the trust of the IoE, and the need to create efficient models capable of addressing dynamic behaviours and in large-scale IoE implementations.

4.7.2 Real-world deployment and scalability considerations

In order to increase the practicality of the offered framework, the HyLite-based trust classification framework may be incorporated into the current smart city or IoE systems via the edge gateways or fog computing devices that track and process the overall behavior of the devices in real-time. Within this type of deployment, the IoE devices (e.g., sensors, public monitoring systems, and smart meters) may send the operational data to the edge nodes in the vicinity, where the HyLite model can then conduct trust evaluation and classification, and hence send the validated data to the central cloud services. Because the proposed HyLite architecture is lightweight and has a small parameter size and memory footprint, the computational overhead is low, and hence it can be used in resource-constrained edge platforms, such as IoE gateways or embedded systems. Overhead on communication is also minimized since the information of the classified trusts or alerts only has to be communicated as opposed to the actual data streams. This can be used in large-scale smart city rollouts to enhance the reliability of the network, authentication of devices and security surveillance, and still be able to do efficient processing at the edge. Further studies in the area of work can be future research on real-time deployment, distributed trust evaluation, and scalability in large IoE ecosystems.

5. CONCLUSION AND FUTURE SCOPE

This study examined how proximity-based trust computation combined with deep learning can improve trust assessment in heterogeneous IoE environments. The experimental results show that integrating spatial interaction patterns with temporal behavior modeling enables more reliable differentiation between trusted, neutral, and untrusted device interactions. The findings indicate that device proximity and interaction duration are key indicators for evaluating trust relationships, and that modeling both local feature patterns and temporal dynamics significantly improves classification performance compared to traditional machine learning approaches. The evaluation across private and public device subsets further highlights that trust behavior varies across device categories, emphasizing the importance of context-aware trust modeling in IoE networks.

Comparative experiments and k-fold cross validation demonstrate that hybrid deep learning architectures can better capture complex behavioral patterns than standalone classifiers. Additionally, the use of explainable AI techniques provided insight into how different features influence trust predictions, improving the transparency of the decision-making process. Overall, the results suggest that combining proximity-based trust estimation with hybrid deep learning models can provide a more effective and interpretable approach for trust management in large-scale IoE systems. Future investigations may focus on refining trust update mechanisms and evaluating the approach on larger and more diverse IoE datasets to further understand its generalization capability.

Future research can further improve the proposed framework by developing fine-grained and adaptive trust

update mechanisms that dynamically adjust trust scores based on evolving device behavior and interaction patterns in IoE environments. In addition, evaluating the HyLite model on larger, more diverse, and better-balanced IoE datasets would help assess its scalability, robustness, and generalization across different network conditions. Further work may also explore integrating additional contextual features and advanced explainable AI techniques to enhance the interpretability and reliability of trust-based decision-making in heterogeneous IoE systems.

REFERENCES

[1] Alsabah, M., Naser, M.A., Albahri, A.S., Albahri, O.S., Alamoodi, A.H., Abdulhussain, S.H., Alzubaidi, L. (2025). A comprehensive review on key technologies toward smart healthcare systems based IoT: Technical aspects, challenges and future directions. Artificial Intelligence Review, 58(11): 343. https://doi.org/10.1007/s10462-025-11342-3

[2] Alserhani, F.M. (2024). Integrating deep learning and metaheuristics algorithms for blockchain-based reassurance data management in the detection of malicious IoT nodes. Peer-to-Peer Networking and Applications, 17(6): 3856-3882. https://doi.org/10.1007/s12083-024-01786-9

[3] Buhnova, B. (2023). Trust management in the Internet of Everything. In Software Architecture. ECSA 2022 Tracks and Workshops, pp. 123-137. https://doi.org/10.1007/978-3-031-36889-9_10

[4] Gill, S.S., Golec, M., Hu, J., Xu, M., et al. (2025). Edge AI: A taxonomy, systematic review and future directions. Cluster Computing, 28(1): 18. https://doi.org/18.10.1007/s10586-024-04686-y

[5] Iftikhar, A., Qureshi, K.N. (2024). Future privacy and trust challenges for IoE networks. In Cybersecurity Vigilance and Security Engineering of Internet of Everything, pp. 193-218. https://doi.org/10.1007/978-3-031-45162-1_12

[6] Mustafa, R., Sarkar, N.I., Mohaghegh, M., Pervez, S. (2024). A cross-layer secure and energy-efficient framework for the Internet of Things: A comprehensive survey. Sensors, 24(22): 7209. https://doi.org/10.3390/s24227209

[7] Safdar, G.A., Bahja, M., Muhammad, M. (2025). Internet of things to internet of humans: A perception. Human-Centric Intelligent Systems, 5(2): 259-268. https://doi.org/10.1007/s44230-025-00100-x

[8] Khatoon, A., Ullah, A., Yasir, M. (2023). Machine learning-based detection and prevention systems for IoE. In Cybersecurity Vigilance and Security Engineering of Internet of Everything, pp. 109-125. https://doi.org/10.1007/978-3-031-45162-1_7

[9] Alghofaili, Y., Rassam, M.A. (2022). A trust management model for IoT devices and services based on the multi-criteria decision-making approach and deep long short-term memory technique. Sensors, 22(2): 634. https://doi.org/10.3390/s22020634

[10] Aaqib, M., Ali, A., Chen, L., Nibouche, O. (2024). A dependable hybrid deep learning model for IoT trust management system. In International Conference on Ubiquitous Computing and Ambient Intelligence, pp. 704-715. https://doi.org/10.1007/978-3-031-77571-0_67

[11] Anwar, R.W., Jabeur, N., Malik, H. (2025). IoTGuard: Enhancing trust management in IoT with long short-term memory and multi-criteria decision-making techniques. SN Computer Science, 6(6): 604. https://doi.org/10.1007/s42979-025-04149-0

[12] Sinha, P., Sahu, D., Prakash, S., Yang, T., Rathore, R.S., Pandey, V.K. (2025). A high performance hybrid LSTM CNN secure architecture for IoT environments using deep learning. Scientific Reports, 15(1): 9684. https://doi.org/10.1038/s41598-025-94500-5

[13] Kilichev, D., Turimov, D., Kim, W. (2024). Next–generation intrusion detection for IoT EVCS: Integrating CNN, LSTM, and GRU models. Mathematics, 12(4): 571. https://doi.org/10.3390/math12040571

[14] Pawar, P.P., Femy, F.F., Rajkumar, N., Jeevitha, S., Bhuvanesh, A., Kumar, D. (2025). Blockchain-enabled cybersecurity for IoT using elliptic curve cryptography and black winged kite model. International Journal of Information Technology, 1-11. https://doi.org/10.1007/s41870-025-02576-z

[15] Karunkuzhali, D., Shaikh, A.A., Suguna, R., Venkatesan, M. (2025). Hybrid lightweight cryptography with attribute-based encryption for secure health monitoring in IOT-wireless body area sensor network. Biomedical Materials & Devices, 4: 2466-2482. https://doi.org/10.1007/s44174-025-00402-5

[16] Ullah, F., Salam, A., Amin, F., Khan, I.A., Ahmed, J., Zaib, S.A., Choi, G.S. (2024). Deep trust: A novel framework for dynamic trust and reputation management in the Internet of Things (IoT)-based networks. IEEE Access, 12: 87407-87419. https://doi.org/10.1109/ACCESS.2024.3409273

[17] Eshmawi, A.A., Aldrees, A., Alharthi, R. (2025). Smart framework for industrial IoT and cloud computing network intrusion detection using a ConvLSTM-based deep learning model. Frontiers in Computer Science, 7: 1622382. https://doi.org/10.3389/fcomp.2025.1622382Al

[18] Ghuraybi, H.A., AlZain, M.A., Soh, B. (2024). Exploring the integration of blockchain technology, physical unclonable function, and machine learning for authentication in cyber-physical systems. Multimedia Tools and Applications, 83(12): 35629-35672. https://doi.org/10.1007/s11042-023-16979-2

[19] Marche, C., Atzori, L., Pilloni, V., Nitti, M. (2020). How to exploit the Social Internet of Things: Query generation model and device profiles' dataset. Computer Networks, 174: 107248. https://doi.org/10.1016/j.comnet.2020.107248

[20] MS, R., Puneetha, Vishwas, Buyya, R., Venugopal, Iyengar, Patnaik. (2020). Trust management for service-oriented SIoT systems. In Proceedings of the 2020 8th International Conference on Information Technology: IoT and Smart City, pp. 216-222. https://doi.org/10.1145/3446999.3447635

[21] Goswami, P., Khan, T., Pathak, V., Alabdultif, A. (2025). Machine learning based dynamic trust estimation framework for securing wireless sensor networks. Scientific Reports, 15(1): 35821. https://doi.org/10.1038/s41598-025-19768-z

[22] Joshi, M., Tiwari, A., Dhabliya, D., Lavate, S.H., Ajani, S.N., Gandhi, Y. (2025). Building AI-driven frameworks for real-time threat detection and mitigation in IoT networks. In 2025 International Conference on Emerging

Smart Computing and Informatics (ESCI), Pune, India, pp. 1-6. https://doi.org/10.1109/ESCI63694.2025.10988310

[23] Mishra, R., Jha, S.K., Prakash, S., Rathore, R.S. (2025). HEXADWSN: Explainable ensemble framework for robust and energy-efficient anomaly detection in WSNs. Future Internet, 17(11): 520. https://doi.org/10.3390/fi17110520