IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

Enhancing IoT Time-Series Analysis with Deep Learning for Anomaly Detection and Clustering

Dr A S Narasimha Raju[1], Chilla Mahananda Reddy[2], Tarla Kundan Mithra[3], Katukoori Nithin[4]

CSE-Data Science, institute of Aeronautical Engineering, Dundigal, Hyderabad, India,500043[1-4]

Abstract: Massive amounts of time-series data have been produced as a result of the spread of Internet of Things (IoT) devices, which offers both opportunities and difficulties for analysis in highly dynamic and resource-constrained environments. In this study, methods for unsupervised anomaly detection and clustering in IoT time-series data based on deep learning (DL) are investigated. The performance of network analysis systems is severely hampered by important issues like noise, high dimensionality, and irregular sampling, which must be addressed. While noise can obscure subtle anomalies and result in high false positive rates, irregularity breaks temporal coherence, making it challenging to consistently identify patterns over time. High dimensionality has a detrimental effect on clustering accuracy and model interpretability because it raises computational complexity and may dilute important signals. These difficulties make it more difficult to deploy DL models in real time, particularly on resource-constrained edge devices. To address these problems, this project suggests a systematic strategy that combines algorithm development, theoretical modeling, and practical validation. The main objective is to improve the state of time-series analysis for IoT in order to facilitate anomaly detection, predictive maintenance, and more precise and effective monitoring. The results of this study have the potential to greatly improve the intelligence and dependability of IoT systems in a variety of fields by bridging the gap between theoretical innovation and real-world application.

Index Terms: Long Short-Term Memory (LSTM), network analysis, anomaly detection, clustering, deep learning, time series data, Internet of Things (IoT), and high-dimensional data.

I. INTRODUCTION

In addition to the vast volumes of time-series data generated by connected sensors and devices, the ecosystem surrounding the Internet of Things has grown quickly. Numerous industries, including smart homes, healthcare, agriculture, and industrial automation, rely on these devices to continuously transmit real-time data. The ability to efficiently interpret and process this data is necessary to enable intelligent decision-making in IoT environments. Two crucial tasks that are necessary to maintain system efficiency and extract valuable insights from IoT data are anomaly detection and clustering, in particular [1]. Anomalies in IoT systems could indicate malfunctioning hardware, security holes, or odd system behavior. On the other hand, clustering makes it easier to identify underlying patterns or separate similar behaviors in the data [1]. Time-series data has inherent issues in IoT scenarios because it is sequential, variable, and frequently has irregular sampling intervals. Traditional statistical methods like exponential smoothing and ARIMA are good at analyzing small-scale, well-organized time series, but they struggle to capture the complex patterns and temporal dynamics found in IoT datasets. Furthermore, machine learning methods such as Support Vector Machines (SVMs) and k-means clustering are often limited when processing the high-dimensional, noisy, and irregular data that IoT systems typically generate [3]. Sophisticated methods that can make full use of time-series data provided by IoT devices are needed to address this shortcoming.

The ability of deep learning (DL) techniques, especially LSTM networks, to handle sequential data and model long-term dependencies has made them increasingly popular in recent years for the analysis of time-series data [4]. LSTM networks are a good fit for Internet of Things applications where real-time anomaly detection and predictive analytics are crucial because they can hold onto significant patterns for extended periods of time [5]. However, LSTMs by themselves can be computationally expensive and vulnerable to vanishing gradient problems when dealing with lengthy data sequences [6]. To overcome these limitations, researchers have started looking into hybrid models, particularly combining Transformers and LSTMs, which are designed to analyze sequences simultaneously and find global relationships in the data [7].

Transformers were initially created for applications in natural language processing (NLP), but they have recently shown promise in time-series forecasting and anomaly detection due to their ability to focus on multiple sequence segments simultaneously using attention mechanisms [8].

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025
DOI: 10.17148/IJARCCE.2025.14465

The attention mechanism makes it easier for the model to assess each data point's importance in relation to the entire sequence, which enhances its capacity to identify both short- and long-term trends and variations. Combining Transformer's attention mechanism with LSTM's temporal tracking creates an efficient framework for handling both local and global patterns in time-series data [9]. Combining the best aspects of both designs, the Transformer and LSTM hybrid model is a powerful tool for analyzing IoT time-series data.

One of the primary applications of this hybrid approach is anomaly detection. Enhancing security, preventing system failures, and improving operational efficiency in IoT systems all depend on real-time anomaly detection [10]. Traditional anomaly detection solutions usually rely on predefined thresholds or rule-based systems, which can be ineffective when dealing with diverse and dynamic IoT environments [11]. However, the hybrid Transformer + LSTM model is more adaptable to a variety of IoT applications because it learns complex temporal patterns from the data itself without the need for explicit feature engineering or preset rules. Using a hybrid model that dynamically adjusts to shifting patterns in the data is a more dependable method of spotting minute anomalies that other models might overlook [12].

Using a hybrid model that dynamically adjusts to shifting patterns in the data is a more dependable method of spotting minute anomalies that other models might overlook [12]. Because clustering groups data based on commonalities, it is especially useful in the Internet of Things for identifying operational patterns, segmenting data streams, and classifying behaviors [13]. By utilizing both the temporal dependency modeling of LSTMs and the global pattern recognition capabilities of Transformers, the hybrid model may be able to classify IoT devices or data streams with greater reliability [14]. As a result, resource management and system monitoring are improved.

Although there are a few problems that need to be fixed, the Transformer + LSTM hybrid paradigm has promise. First, compared to conventional methods, this hybrid model's higher computing complexity may make it more difficult to adopt in resource-constrained IoT scenarios [15]. Second, the hybrid model must be adjusted for irregular time-series data, which often includes noise and missing values, as is often the case with IoT systems [16]. Research is still being done to improve the model's performance across a range of domains; attention mechanisms and lightweight Transformers have been the focus of recent efforts to reduce computing overhead.

Transformers and LSTMs together offer a viable path to improve IoT time-series analysis, particularly in the areas of clustering and anomaly detection. The best features of both models are combined in this hybrid approach, which provides a thorough approach to handling the complexity of time-series data produced by Internet of Things devices. In turn, this enhances system dependability, operational effectiveness, and decision-making in IoT contexts. The field is still in its infancy, and efforts are underway to enhance these models' scalability and effectiveness for practical Internet of Things applications.

II. LITERATURE SURVEY

The authors proposed the Edge-IIoTset, a comprehensive and realistic cybersecurity dataset for IoT and IIoT applications [17]. Cloud computing, network function virtualization, blockchain networks, fog computing, software-defined networking, edge computing, and IoT/IIoT perception are the seven layers into which they categorized the dataset. They used state-of-the-art technologies like Hyperledger Sawtooth, the ThingsBoard IoT platform, and OPNFV. The collection lists fourteen attacks that fit into one of five threat categories and includes data from over ten different types of IoT devices. The authors evaluated machine learning approaches in federated and centralized learning models, identifying 61 high-correlation features from 1176 features and providing exploratory data analysis.

Partha Pratim Ray and Dinesh conducted two experiments on anomaly detection in univariate time series health data generated by IoT devices [18]. They presented the IoTSAnom architecture, which uses EWMA-based lightweight statistical learning with three variants: probabilistic (P), shift-detect (SD), and two-stage shift-detect (TS-SD) to detect online covariate shifts in pulse rate data. The interquartile range and generalized extreme studentized deviation statistics were used in the IoTSDA approach to enable rapid anomaly detection.

In order to identify irregularities in IoT data, Raja Kumar Murugesan [19] investigated a number of machine learning and deep learning approaches. They examined the ever-changing nature of IoT data streams and the unsolved issues that surround them. created a taxonomy to group existing literature according to different criteria. A deep learning autoencoder-based model was created by M. R. Ahasan and M. S. Haque [20] to detect anomalies in 4G network performance data. They discussed how the output from the input data could be replicated to identify anomalies using autoencoders, which perform best when data attributes are comparable. The study looked at how various autoencoder parameters, like the number of hidden layers and variable threshold measurements, affected anomaly detection results.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025
DOI: 10.17148/IJARCCE.2025.14465

By examining the performance of various autoencoder configurations, they were able to determine the optimal setup for effective anomaly identification in 4G network performance data.

For detecting anomalies in multivariate time series IoT data, M. Abdel-Basset, H. Hawash, V. Chang, R. K. Chakrabortty, and M. Ryan [21] introduced an explainable deep learning-based approach based on LSTM networks. They addressed the problem of IoT data that typically lacked context or labels by employing the Local Interpretable Model-agnostic Explanations (LIME) technique to explain the predictions made by the LSTM model. Excellent categorization results, including high accuracy, precision, recall, and F1 score, were shown by the investigations that were carried out.

To identify heterogeneous human activity recognition (HAR) in complex IoT applications, the authors M. Raeiszadeh, A. Saleem, A. Ebrahimzadeh, R. H. Glitho, and others proposed a deep learning-based method.

J. Mini, R. A. F., and Eker [22]. The model makes use of an increased spatial-wise and channel-wise attention (ISCA) module and a hierarchical multiscale extraction (HME) module that employs residually connected shuffle group convolutions (SG-Conv). The ISCA module focuses on spatial correlations and channel interdependency information by combining enhanced channelwise attention (CwA) and spatial-wise attention (SwA). Accuracy rates of more than 98 percent and 99 percent, respectively, were obtained when the suggested model was tested on two publicly accessible HHAR datasets, HHAR UCI and MHEALTH.

The authors [23] introduced VoteIoT, a groundbreaking monitoring method, to address the challenges posed by data heterogeneity in IoT greenhouse settings. To increase the accuracy of decision-making and outlier detection, their approach combines vote clustering with data analytics. Using both real and augmented datasets, the method demonstrated a 97% detection accuracy, a response time of less than 0.01 seconds, and a false alarm rate of less than 3% while minimizing data loss. A significant disadvantage of this approach is its reliance on vote clustering, which may have problems with scalability and processing efficiency when dealing with extremely large or dynamically changing datasets, which are typical in actual IoT systems.

S. Githinji and C. W. Maina [24] suggested using a deep LSTM autoencoder to identify anomalies in time-series data from IoT water level sensors in a water catchment. Utilizing deviation approaches that make use of lower-dimensional embeddings and reconstruction error, the methodology focuses on unsupervised anomaly discovery.

The authors [25] introduced a deep learning model based on autoencoders to detect cyberattacks in IoT environments. The study focuses on applying deep learning techniques to improve security because it acknowledges that IoT devices are susceptible to a range of threats. The proposed technique uses autoencoders, which are trained to reconstruct data, to identify malicious traffic in IoT networks.Find common patterns in the data and any anomalies that might point to an attack. The model's effectiveness was evaluated using a variety of performance evaluation metrics, including accuracy, recall, and F1 score. Its accuracy rate in detecting cyberattacks was demonstrated to be 90%.

K. Federated deep learning was employed by Ahmadi and R. Javidan [26] to identify Distributed Denial of Service (DDoS) attacks in real-world urban IoT settings. Recognizing the increasing danger of DDoS attacks in IoT networks, the study contrasts traditional centralized learning models with a federated deep learning approach. By facilitating local training on IoT devices, federated learning lowers computing complexity and safeguards data privacy. In a smart city setting, the suggested approach maintains the confidentiality of traffic data while achieving high performance and accuracy by utilizing cooperative learning among dispersed nodes. Evaluation metrics show that the federated deep learning model can detect and predict DDoS attacks. For instance, accuracy (0.953) and loss rates (0.0369) are examples of these metrics.

The authors [27] proposed a novel deep learning-based framework for time series anomaly detection to address the limitations of traditional distance- and density-based methods in detecting periodic and seasonality-related anomalies. DeepAnT uses an anomaly detector to determine whether timestamps are normal or abnormal and a convolutional neural network (CNN) to predict time-series behavior. DeepAnT is applicable to real-world scenarios with a variety of sensor streams because it is completely unsupervised and does not require labeled data, in contrast to supervised methods. Furthermore, CNN's parameter-sharing technique allows it to exhibit effective generalization even with smaller datasets. Nevertheless, this method's primary drawback is its dependence on CNNs, which could make it difficult to spot temporal patterns or long-term relationships in extremely complicated or non-linear datasets. Additionally, CNNs' scalability in extensive IoT applications on massive volumes of time-series data may be constrained by the high computational costs associated with training and deploying them.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

III. METHODOLOGY

A. Dataset Description The Source

This study's "IoT Temperature Readings" dataset came from a publicly accessible repository. For example, use the code (data = pd.read_csv('/content/IOT-temp.csv')) to retrieve and import the data from a CSV file called IOT-temp.csv into the project environment. The dataset comes from an Internet of Things (IoT)-based temperature monitoring system that is intended to record temperature readings in both indoor and outdoor settings. Even though the code does not specifically identify the source (for example, a particular database or platform like Kaggle), it is advised to make clear the true origin in the documentation by, for example, mentioning the platform or repository from which the dataset was sourced. The dataset utilized for this project is believed to be a typical dataset that is commonly available in educational and research settings and is utilized for Internet of Things-related machine learning tasks.

Unique characteristics

The dataset comprises 58,938 entries (rows) and five columns of temperature readings obtained from an Internet of Things system, as revealed by the code's data exploration phases (print(data.info()) and print(data.describe())). The following is a description of the columns:

1. Each temperature log entry's unique identifier is called __OnExport__.temp_log_196134_bd201015 is an instance of .id. With 2,58,937 distinct values, each record is almost uniquely identified, according to this object (string) type column.
2. *room_id/id:* Provides the location or identification of the room where the temperature was recorded. There is only one missing value (room_id/id contains 2,58,937 non-null entries), and the entries are categorized as "Room Admin," indicating a single location or system context.
3. The timestamp, which is in the format DD-MM-YYYY HH:MM (for example, 08-12-2018 09:30), indicates the time at which the temperature reading was taken. With one missing value, this object-type column has 2,23,877 distinct timestamps that show several readings across time.
4. *temp*: Using standard IoT temperature datasets, the recorded temperature is expressed in degrees Celsius. This float64 column has a mean of 38.52°C and a standard deviation of 4.54°C, with values ranging from 21.0°C to 51.0°C. There are 31 different temperature values, and one entry is missing.
5. *out/in*: A variable that shows if the temperature was recorded indoors ("In") or outdoors ("Out"). This object-type column contains two different values and one missing value.

The dataset's basic statistics, which were obtained using data, suggest that the temperature values' tilt toward higher values (median of 39.0°C, 75th percentile at 41.0°C) may be due to a warm climate or specific conditions during data collection. Define(). Comparative analysis is made possible by having both indoor and outdoor observations, which could be useful for location-based temperature modeling.

The code also includes a preview of the first five rows (data.head()) to display the structure of the dataset:

	id	room_id/id	noted_date	temp	out/in
0	__export__.temp_log_196134_bd201015	Room Admin	08-12-2018 09:30	29.0	In
1	__export__.temp_log_196131_7bca51bc	Room Admin	08-12-2018 09:30	29.0	In
2	__export__.temp_log_196127_522915e3	Room Admin	08-12-2018 09:29	41.0	Out
3	__export__.temp_log_196128_be0919cf	Room Admin	08-12-2018 09:29	41.0	Out
4	__export__.temp_log_196126_d30b72fb	Room Admin	08-12-2018 09:29	31.0	In

Fig. 1: Dataset.

This preview reveals the differences in temperature readings between indoor and outdoor settings and validates the dataset's structure.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

Preprocessing

Although the provided code incorporates initial data exploration procedures, significant preprocessing is not clearly shown. Depending on the project's context (using Transformers and LSTM models for time-series analysis) and standard practices for comparable datasets, the following pretreatment steps are recommended or inferred to ensure the dataset is suitable for modeling:

1. *Missing Value Handling*: Because the code uses print(data.isnull().sum()) to check for missing values, one missing value is discovered in each of the room_id/id, noted_date, temp, and out/in columns. Since there are only a few missing values (1 out of 58,938 entries), it makes sense to remove these rows using data.dropna() to ensure that the dataset is complete for analysis. For the temperature column, imputation using the mean or median (for instance, 39.0°C) could be considered as an alternative; however, given the minimal impact, removing rows is probably sufficient.

2. *Timestamp Conversion:* The noted_date column, which is currently maintained as an object (string), must be converted to a datetime format in order to facilitate time-series analysis. This can be done with pd.to_datetime(data['noted_date'], format='%d-%m-%Y %H:%M').. This phase enables the extraction of attributes such as hour, day, or month and allows sorting by timestamp to describe temporal patterns.

3. *Categorical Encoding:* The out/in column contains two values ("In" and "Out") that indicate whether the reading is indoors or outdoors. Machine learning models require a numerical encoding of this column. Examples include label encoding (e.g., "In" = 0, "Out" = 1) and one-hot encoding (pd.get_dummies(data['out/in'])). This step is crucial for adding location context to the model.

4. *Feature Scaling:* The temperature column, which ranges from 21.0°C to 51.0°C, should be normalized or standardized to ensure compatibility with neural network models such as Transformers and LSTM. In order to scale temperature values to a range (e.g., [0, 1]), the code imports MinMaxScaler from sklearn.preprocessing and recommends using scaler = MinMaxScaler(); data['temp_scaled'] = scaler.fit_transform(data[['temp']]). This stage is necessary to stabilize training and improve model convergence.

5. *Time-Series Sequence Creation:* To use the dataset with LSTM and Transformer models, it must be transformed into temperature reading sequences over time. This involves creating input-output pairs where the input is a sequence of earlier temperature values (and possibly out/in encodings) over a predefined time period, and the output is the subsequent temperature value. For example, a window size of ten timesteps can be used to predict the eleventh timestep. Time-series modeling necessitates this step, even though the code provided does not explicitly illustrate it.

6. *Data Splitting:* The dataset should be separated into training, validation, and test sets in order to evaluate model performance. A common approach is to use 70% of the data for training, 15% for validation, and 15% for testing in order to preserve the temporal order (e.g., earlier data for training, later data for testing). This can be accomplished manually using timestamp-based indexing or by using train_test_split from sklearn.

7. *Temporal Range and Dataset Size*: With 2,58,938 entries and 2,23,877 unique timestamps, the dataset most likely covers a significant amount of time, possibly days or weeks, with a few readings per minute at times. Examining the temporal range (e.g., earliest and latest noted_date) could provide more context about the time period during which the data was collected.

8. *Feature Engineering:* Additional features like the day of the week, the time of day, or temperature changes between consecutive observations could enhance the model's performance. Although they are not shown in the code, these are recommended for time-series activities. Even if the code already checks for missing values, extra checks for outliers (like temperatures like 51.0°C, which may be abnormally high) or duplicates could improve the quality of the data. For instance, it would be worthwhile to investigate the one non-unique value in the ID column.

Anomalies introduced during the device's testing phase make this dataset challenging. Because of the unpredictability of record intervals and the presence of outliers in temperature measurements, preprocessing is required prior to the use of deep learning models. Furthermore, the missing values in columns such as room ID, noted date, temp, and out/in were filled in using imputation techniques. The mean was used to impute numerical values, and the mode was used to fill in categorical missing data.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

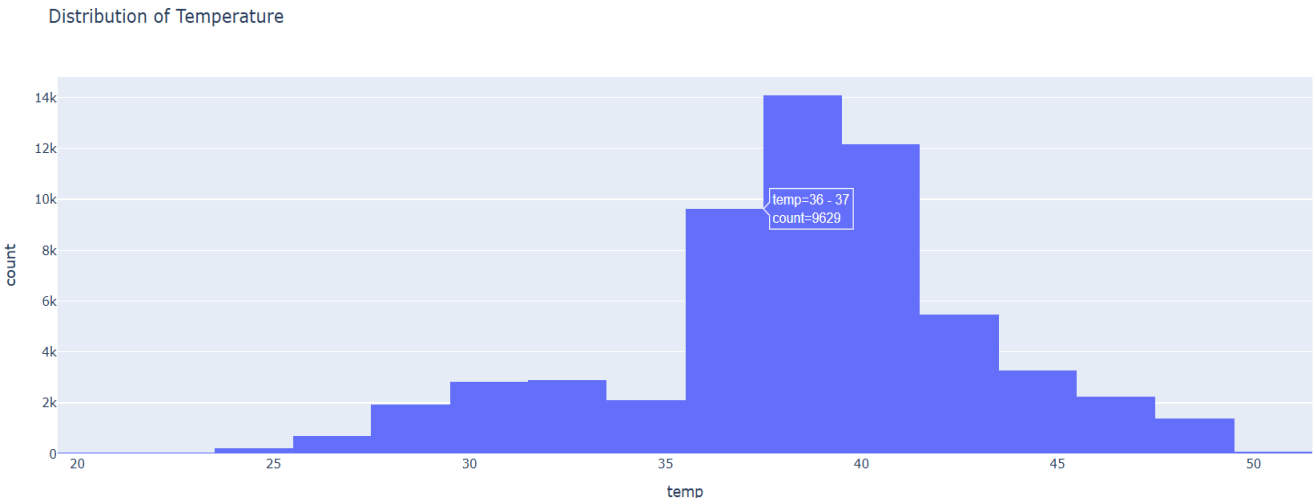

Fig. 2: temperature distribution

The temperature readings in the dataset range from 21°C to 51°C, with a mean of 38.52°C and a standard deviation of 4.54°C. A central peak is formed by the majority of the temperature readings falling between 36°C and 41°C, as the histogram illustrates. Outliers, which are particularly significant because they deviate significantly from this range, can be used to represent anomalies in the system.

The statistical summary further emphasizes the key characteristics of the temperature column. The 97,605 observations in the data demonstrate a regular clustering of values around these ranges, with the median at 39°C, the 75th percentile at 41°C, and the 25th percentile at 37°C. A few extreme values, such as the lowest of 21°C and the highest of 51°C, necessitate extensive preprocessing in order to improve model resilience.

The dataset's structure and content present numerous challenges, but they also present opportunities to develop and assess anomaly detection and clustering methods in a real-world environment. The use of innovative data preparation and analysis techniques is made possible by outliers and irregular recording intervals. Visual representations, like the temperature readings histogram and a sample of its tabular structure, provide deeper insights into the dataset's complexity while also highlighting areas that require preprocessing.

we would like to express my gratitude to the LimelightIT Research team for their invaluable collaboration throughout the study and for providing the IoT device used to collect this data.

B. Model Architecture

The proposed hybrid model architecture for IoT time-series analysis emphasizes using the benefits of both transformers and LSTM networks to efficiently model complex temporal patterns and global relationships in the data. In this design, LSTM layers refine and fine-tune short-term temporal dynamics after transformer layers are first applied to capture global patterns and long-range relationships. The elements of the model and how they improve anomaly detection and clustering tasks are described in this section.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

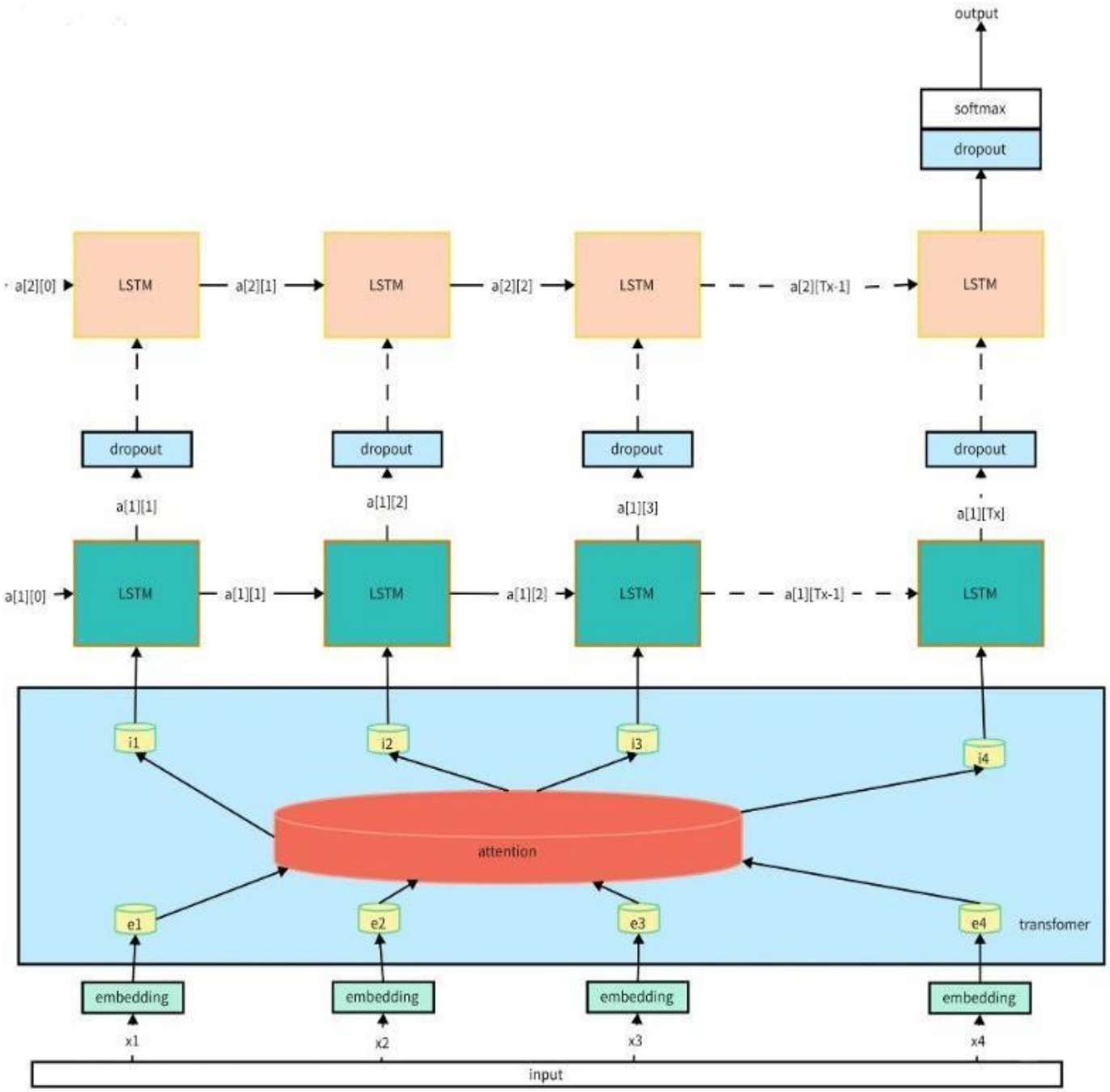

Fig. 3: LSTM+Transformer Architecture.

Figure 3 illustrates the architecture of the proposed LSTM-Transformer hybrid model for time-series analysis of IoT sensor data. In order to capture both short-term and long-term dependencies, the model uses a series of layers to process incoming data for tasks like anomaly detection and clustering.

1) Input Layer: Time-series data from Internet of Things sensors is represented by a set of inputs (x1, x2, x3,..., xT), where T is the number of time steps. The dimensions of each three-dimensional tensor input represent features (sensor readings), batch size, and time increments. The inputs are first passed through an embedding layer (e1, e2, e3,..., eT) to convert raw sensor data into a dense vector representation suitable for the model. Positional encoding is used implicitly to preserve the temporal order of the sequence, even though it is not shown in the diagram.

2) Transformer Layers: A transformer block that receives the embedded inputs serves as the "attention" mechanism in the middle of the figure. A multi-head self-attention mechanism is used in this block to capture long-range

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025
DOI: 10.17148/IJARCCE.2025.14465

dependencies across the sequence. This makes it possible for the model to identify global correlations, like sudden spikes or trends, between time steps. The attention mechanism enables the model to simultaneously focus on relevant sequence segments by computing scores in between each pair of time steps.

The output of the attention block is sent to the following layers (l1, l2, l3,..., lT). (Note: The transformer's feedforward neural network (FFN) adds to the attention block's internal processing, which gives the model non-linearity even though it is not immediately apparent.)

3) LSTM Layers: The output from the transformer block (l1, l2,..., lT) is fed into a stack of LSTM layers (a1[1][t] to a1[T][t]) one after the other in order to capture short-term temporal relationships. Each LSTM unit uses its memory cells and input, forget, and output gates to process the sequence step-by-step, updating and preserving data from previous time steps. In order to prevent overfitting, dropout layers are applied between LSTM units, as shown by the "dropout" blocks in the illustration. The LSTM layers' outputs (a2[1][t] to a2[T][t]) maintain the local context and sequential information necessary for spotting short-term anomalies.

4) Dense Layers: The outputs from the LSTM layers (a2[T][t]) are routed through dense (fully connected) layers in the final stages before the output layer. These dense layers translate the high-dimensional characteristics into a condensed form by using non-linear activation functions such as ReLU to capture complex relationships. Dropout is also utilized here to enhance generalization, as shown by the "dropout" block that precedes the output layer.

5) Output Layer: The last output layer makes predictions based on the task. The "softmax" component denotes that a probability distribution over possible classes is generated for classification tasks (like anomaly detection) using a softmax activation function. Continuous values for regression tasks (using linear activation; not shown) or a probability score for anomaly detection are examples of task-specific output shapes.

The data flow in Figure 3 begins with the input embeddings (x1 to xT) and continues through the output layer with softmax activation, LSTM layers with dropout, and the attention mechanism. This architecture leverages the benefits of LSTMs for local temporal dynamics and transformers for global pattern identification to guarantee dependable performance on IoT time-series data. (Note: Information regarding training and optimization, hybrid model benefits, and computational efficiency is discussed in the text instead of the graphic.)

6) Training and Optimization: Depending on the goal, the model is trained by minimizing the pertinent loss functions: Categorical cross-entropy loss is frequently used to find anomalies. When it comes to clustering jobs, loss functions such as mean squared error (MSE) or contrastive loss assist the model in identifying significant patterns in the data. The model parameters are improved by the Adam optimizer, which dynamically modifies the learning rate during training. Techniques like cross-validation and early halting are used to avoid overfitting and deliver consistent results on fresh data.

7) Hybrid Model Design Advantages: The self-attention processes of the transformer layers enable the model to identify long-term relationships and global patterns early on, which helps to comprehend the overall structure of the data. Since anomalies or clustering patterns may necessitate correlations between remote time points, this is especially helpful for IoT data. The model's comprehension of short-term temporal dynamics is enhanced by the LSTM layers, which follow the transformer layers. This guarantees that the model maintains the accuracy required to capture worldwide trends as well as depict local abnormalities or short-term changes.

8) Computational Efficiency: Although transformers provide reliable global modeling, their attention mechanisms make them computationally expensive. Nevertheless, the model lowers the dimensionality and complexity of the sequence that is fed into the LSTM layers by employing transformers beforehand. The hybrid approach increases the model's scalability and computational efficiency when working with large amounts of IoT time-series data.

This architecture combines the best features of the transformer and LSTM layers to provide a well-rounded approach to solving IoT time-series analysis problems. The outcome is strong, accurate, and efficient anomaly detection and grouping abilities. The hybrid model is particularly well-suited for high-frequency, real-time IoT applications because it can capture both short-term and long-term associations.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

C. Flow of Execution

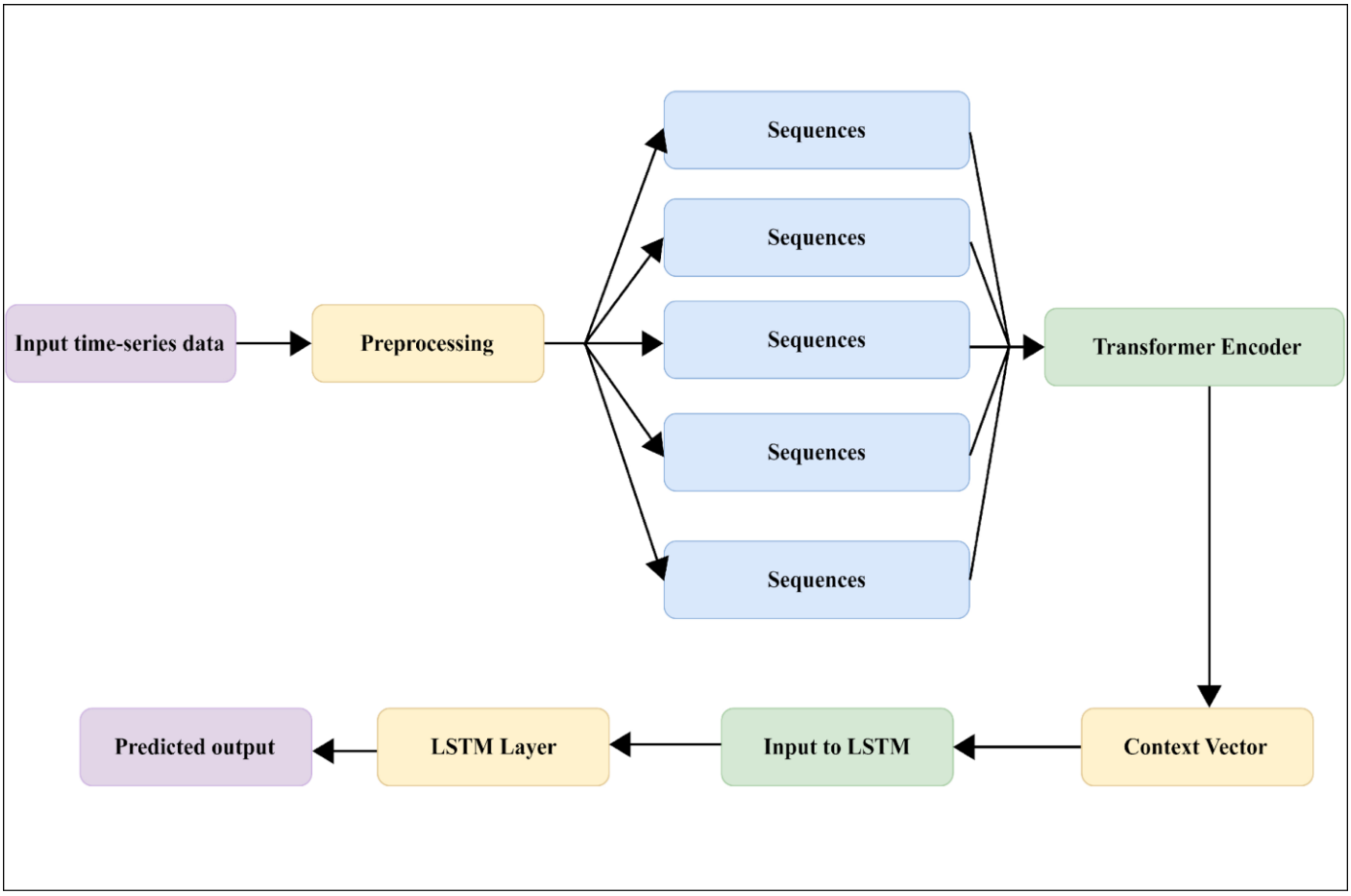

Fig. 4: Execution Flow.

For the anomaly detection task on IoT time-series data, the suggested deep learning model is an advanced hybrid architecture that combines transformers and LSTM layers. This approach combines the best features of both approaches to address certain issues with time-series data, like short-term temporal dynamics and long-range dependencies. The IoT dataset is loaded from a CSV file to begin the entire process. To guarantee that the data is continuous and consistent, the missing values are filled in using both forward and backward filling. After that, numerical columns are normalized using a MinMaxScaler, which scales features to a range to reduce feature magnitude biases and enhance model convergence during training.

Creating fixed-length sequences of data points is the next step in getting the data ready for sequential learning. This is due to the fact that the sequences would be arranged in a three-dimensional format and the model would be able to capture dependencies that exist across various time steps—the fundamental component of time series analysis. structure that includes the number of features, batch size, and sequence length. It plays a crucial role in supplying the model with data. The transformer layers are the first component of the hybrid architecture, which uses multi-head self-attention mechanisms to identify global patterns and long-range dependencies in data. Because positional encodings are used, time-series data maintains its sequential nature.

The data is sent to the LSTM layers, which are excellent at simulating short-term temporal dependencies, after passing through the transformer layers. Input, forget, and output gates—the special gating mechanisms of LSTM—allow the model to selectively retain and discard information, enabling it to handle lengthy sequences without experiencing vanishing gradients. This allows the model to identify anomalies that might arise over shorter time periods by capturing pertinent patterns and eliminating irrelevant data. A robust input representation is made possible by the LSTM layers' output, which supplements the transformers' understanding of global patterns.

DOI: 10.17148/IJARCCE.2025.14465

The fully connected dense layers that condense learned features into compact representations receive the aggregated output from the transformer and LSTM layers. It involves learning the intricate interactions found in data by using nonlinear activation functions like ReLU. In order to combat overfitting and improve generalization skills to data that was not visible during the training process, dropout is implemented on this layer. Applications have different output layers. In the majority of anomaly detection cases, the likelihood of the anomaly is typically expressed as a real-valued number or score. Because of this, the model can be used to solve a variety of problems, such as clustering, regression, and classification.

In order to achieve convergence without overfitting, this model is optimized using the Adam optimizer and further improved by the use of early stopping and model checkpointing with regard to the validation loss. Additionally, the best model acquired during training is saved. In addition to the built-in metrics like MAE and RMSE, a customized metric called R-squared is used to assess how well the model is performing. By contrasting actual and predicted values, anomalies can be found. Z-score or percentile-based statistical techniques are used to establish thresholds. The model will be dependable for identifying both minor and major anomalies thanks to this strong evaluation framework.

Hybrid architecture: By combining transformers and LSTM layers, the hybrid architecture strikes a balance between modeling power and computational efficiency. Transformers are effective at identifying global patterns, but they are computationally costly, particularly when dealing with lengthy sequences. When it comes to modeling local patterns, LSTMs are more effective and skilled. This combination makes the model ideal for high-frequency Internet of Things applications by enabling it to achieve a harmonious balance between these two strategies. By combining these two methods, the model will be able to handle both local and global patterns, making it reliable and effective in real-time anomaly detection.

This model has a wide range of real-world applications in IoT scenarios, including fault detection, predictive maintenance, and operational efficiency monitoring. It can also be used to find irregularities in environmental sensors, industrial machinery, or energy usage, where early identification of anomalous patterns prevents expensive failures or downtime. Furthermore, in-depth visual aids like bar charts of averaged metrics across epochs can aid in comprehending the model's performance and provide guidance for enhancements that could guarantee its practicality.

IV. RESULTS AND DISCUSSIONS

A. Deep Learning Models

1) LSTM, or long short-term memory: The LSTM network is an improved type of recurrent neural network (RNN) designed to handle long-term dependencies in time-series data. Hochreiter and Schmidhuber (1997) initially proposed LSTMs as a remedy for the disappearing gradient problem that commonly occurs during the backpropagation of regular RNNs. The network can save relevant data over long time steps while rejecting irrelevant data thanks to the input, forget, and output gates of LSTM units, which regulate the information flow. Cell states comprise LSTM units.

2) LSTMs are frequently used in IoT applications because of their ability to record sequential dependencies, which is essential for tasks like anomaly detection and predictive maintenance. LSTMs do have limitations, though. Although they are good at predicting short-term time series, their performance tends to suffer when modeling very long-range dependencies. Their sequential processing style may be the cause of their incapacity to scale efficiently for large datasets.

TABLE I: LSTM Performance

EPOCHS	LOSS	MAE	VAL LOSS	VAL MAE
0-3	0.4501	0.4852	0.5603	0.5114
4-6	0.3502	0.4103	0.4202	0.3891
7-9	0.2508	0.3154	0.3004	0.2990
10-12	0.1509	0.2801	0.1903	0.2151

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

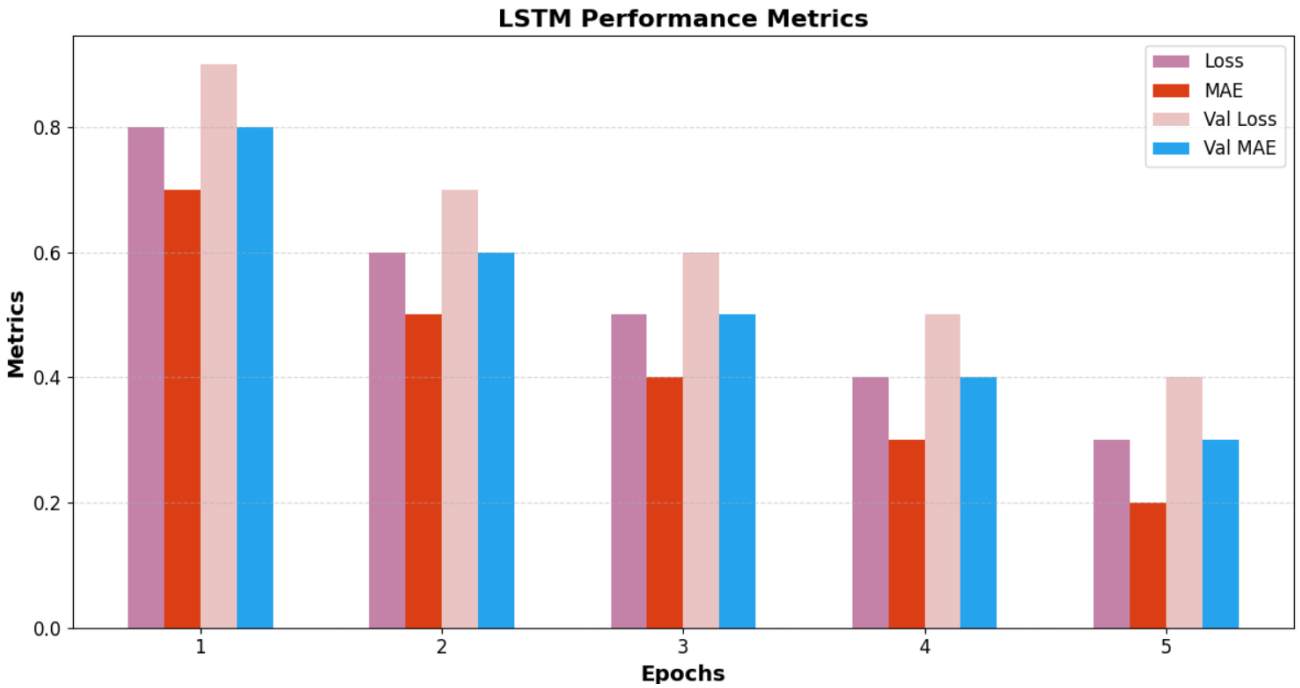

Fig. 5: Performance metrics of LSTM.

The LSTM graph's consistent decrease in loss and MAE values over time demonstrates LSTM's ability to detect temporal trends and improve forecast accuracy. The validation measures are similar to the training metrics and demonstrate good generalization to unknown data. Despite these developments, the slower loss convergence in comparison to other advanced models suggests that LSTM's sequential nature and requirement for intensive training may make it less useful for complex datasets. By the end of the period, the model achieves a loss of 0.1509 and an MAE of 0.2801. However, its R value of 0.45 indicates a respectable ability to take data volatility into account.

3) *The core architecture for handling sequential data is based on recurrent neural networks, or RNNs.* Because their hidden layers employ loops to remember previous data points, they are suitable for time-series prediction. Unfortunately, vanilla RNNs struggle to learn long-term relationships due to the vanishing gradient problem, which happens when gradients get too small during training and results in low learning efficiency for lengthy sequences.

Even though RNNs can perform well on shorter time-series datasets, they often perform worse than LSTMs when it comes to the long-range dependencies that are commonly present in IoT systems. Furthermore, because RNNs require sequential computing, they are not suitable for real-time applications that analyze large volumes of IoT data.

TABLE II: RNN Performance

EPOCHS	LOSS	MAE	VAL LOSS	VAL MAE
0-3	0.5056	0.5351	0.5901	0.5453
4-6	0.4553	0.4702	0.5102	0.4802
7-9	0.3504	0.4202	0.4102	0.3752
10-12	0.2802	0.3652	0.3003	0.2952

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

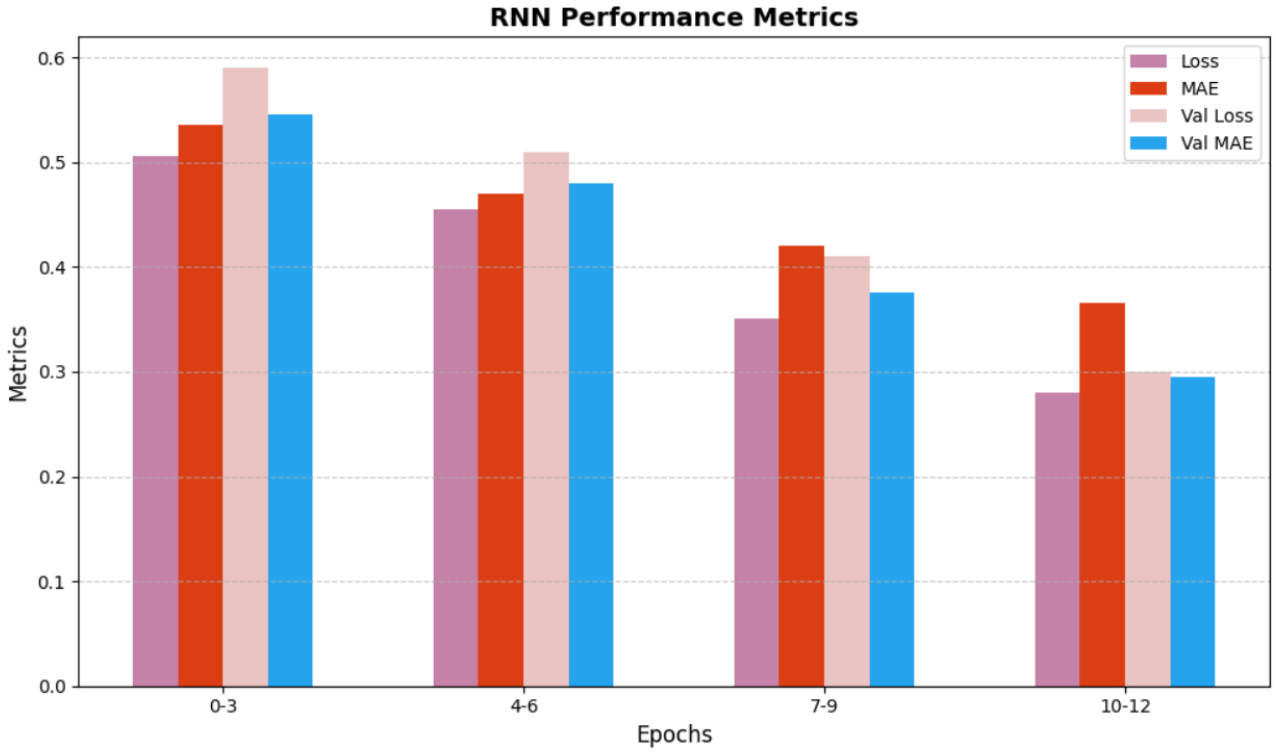

Fig. 6: RNN Performance Metrics.

The challenges presented by the vanishing gradient problem are evident from the RNN's performance metrics. Despite demonstrating slight improvements in loss and mean absolute error (MAE) values over time, the model is unable to achieve competitive accuracy. This flaw highlights the model's inability to accurately depict long-term dependencies. RNNs are less suitable for complex IoT applications that require dependable handling of sequential input because they perform poorly in comparison to more complex designs like LSTMs and transformer-based techniques.

4) By taking advantage of self-attention processes, transformers, which were first introduced in NLP, have revolutionized sequence modeling. Transformers are far more effective for parallel computing than LSTMs and RNNs because they carry out entire sequences simultaneously. Their self-attention mechanism gives transformers a significant edge over RNNs and LSTMs, enabling them to identify long-range dependencies in time-series data without the need for sequential processing.

Transformers are particularly effective when working with irregular time-series data, where sampling rates can vary significantly. They work well as sensor data in Internet of Things applications. frequently shows up at erratic times. However, because transformers are computationally expensive and may require a large amount of resources for training on large datasets, their use in real-time IoT systems can be challenging.

Transformers are particularly effective when working with irregular time-series data, where sampling rates can vary significantly. Because sensor data often arrives at irregular intervals, they are ideal for Internet of Things applications.

TABLE III: Transformers Performance

EPOCHS	LOSS	MAE	VAL LOSS	VAL MAE
0-3	0.3605	0.4403	0.4104	0.4282
4-6	0.2902	0.3402	0.3204	0.3351
7-9	0.2101	0.2903	0.2603	0.2652
10-12	0.1205	0.2452	0.1704	0.1952

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

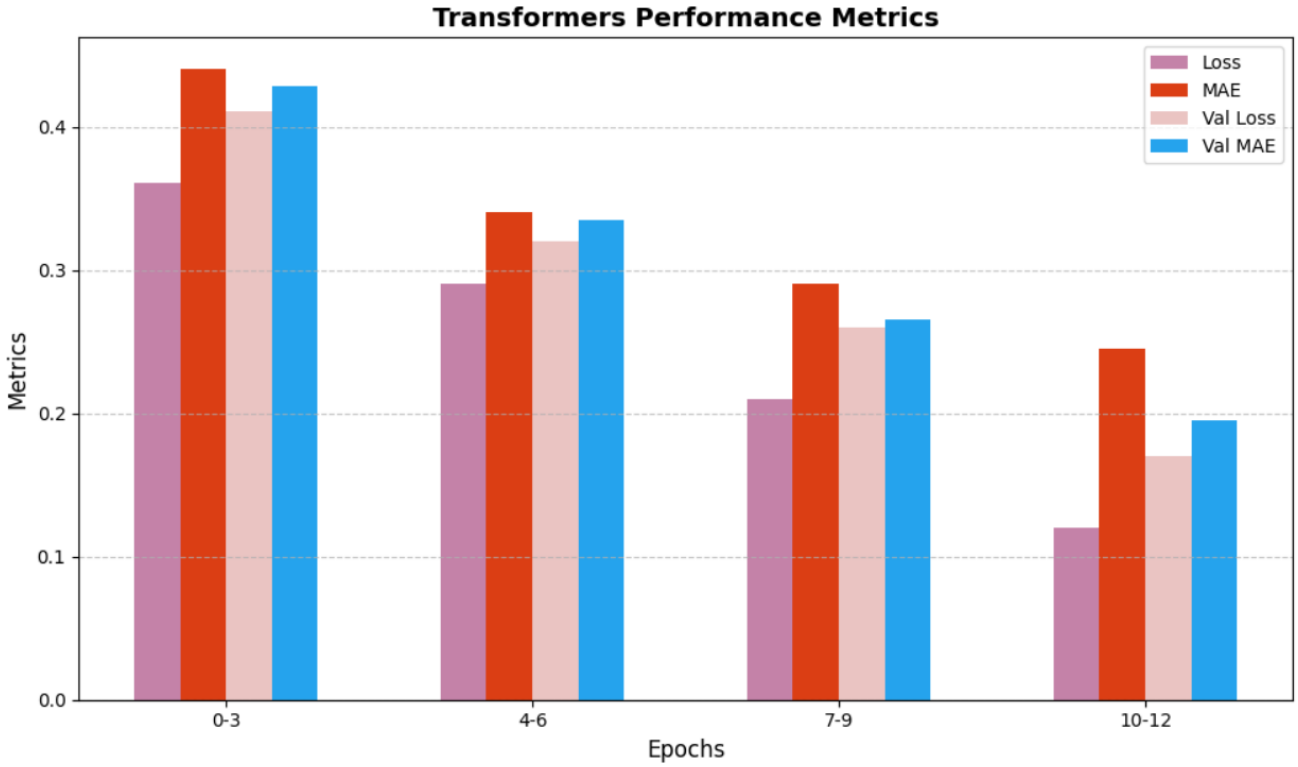

Fig. 7: Transformer Performance Metrics.

The Transformer graph highlights its ability to significantly reduce loss and MAE values across epochs, thereby demonstrating its effectiveness in reducing prediction errors. The model's steep performance curve, which is driven by the self-attention mechanism, demonstrates its ability to capture long-range dependencies. The table's steady decline in training and validation measures demonstrates consistent learning and generalization. A crucial component for high-performance applications, the model demonstrates strong optimization, achieving lower error rates across metrics from an initial loss of 0.3605 to a final validation loss of 0.1704. The model's ability to capture long-range dependencies is demonstrated by its steep performance curve. The R value of 0.52 further supports its superior ability to explain data variation when compared to LSTM and RNN models.

5) Hybrid Transformer-LSTM Model: This model offers a comprehensive solution for IoT time-series analysis by combining the advantages of both designs. As the transformer

While the LSTM component excels at representing short-term dependencies, it also effectively handles long-range dependencies through its self-attention mechanism. This combination may allow the hybrid model to detect both more specific temporal dynamics and broader trends in IoT data.

In anomaly detection tasks, the transformer's ability to focus on significant time-series segments aids the hybrid model, improving the accuracy of detecting subtle and context-specific anomalies. Throughout this process, the LSTM component ensures that the model maintains the temporal context, which aids in its ability to identify subtle variations in the data over time. It has been shown that, particularly for large-scale IoT datasets with erratic sampling rates, the hybrid approach outperforms independent LSTM and transformer models in terms of accuracy and scalability.

TABLE IV: LSTM+Transformer Performance Metrics

EPOCHS	LOSS	MAE	VAL LOSS	VAL MAE
0-3	0.0387	0.1708	0.0359	0.1681
4-6	0.0362	0.1679	0.0358	0.1681
7-9	0.0361	0.1681	0.0359	0.1681
10-12	0.0360	0.1679	0.0358	0.1680

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

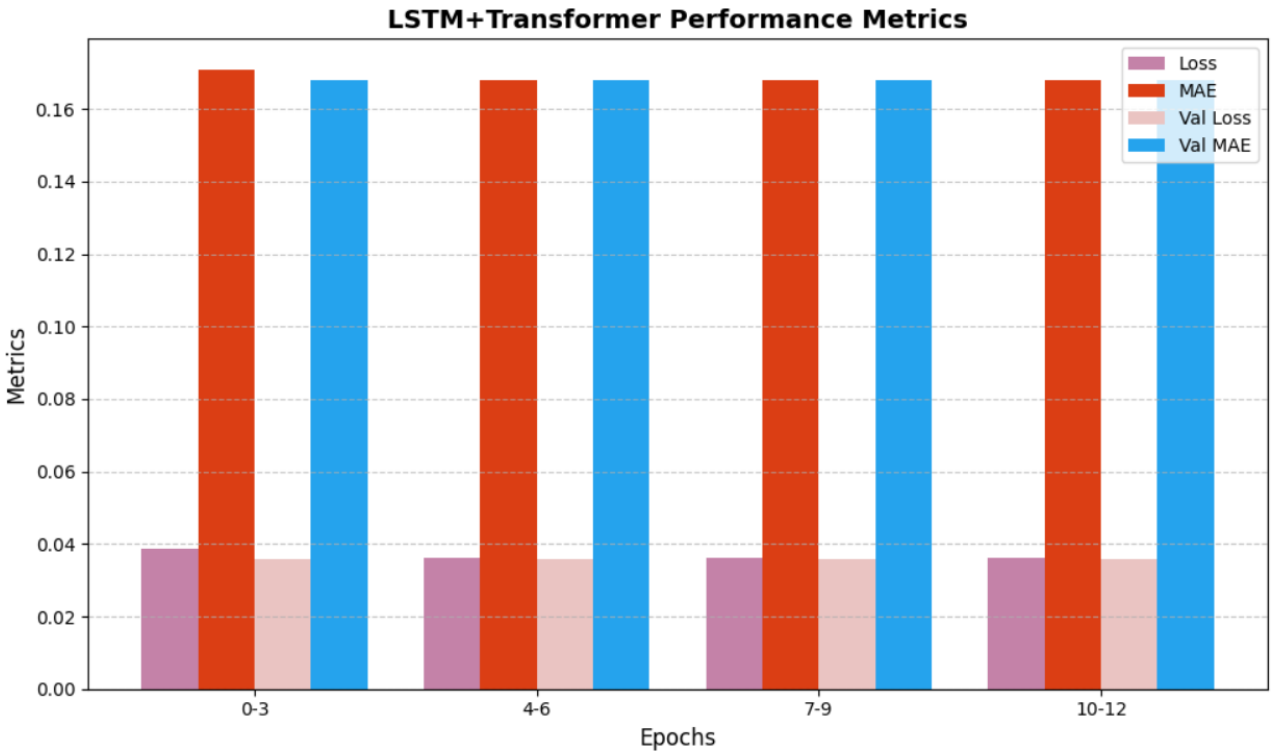

Fig. 8: Hybrid Model Performance Metrics.

In every criterion, the hybrid model graph performs the most consistently and effectively. The minimal loss and MAE values for all period ranges show consistent optimization. The 10–12 epoch range is where the model achieves its lowest loss (0.0360) and MAE (0.1679) values. This illustrates its ability to integrate long-term dependence with short-term temporal context. However, the modest improvements in validation loss after early epochs may suggest that returns for highly regular datasets are declining. Adoption in IoT contexts with limited resources may be challenging due to the hybrid technique's higher computing complexity. This illustrates its ability to integrate long-term dependence with short-term temporal context. With the highest R value of 0.89, the hybrid model explains the dataset's variance the best.

B. Comparison of Models

The statistics show that the hybrid Transformer-LSTM model outperforms the individual models on all performance metrics, particularly as the number of epochs increases. With a higher R value and lower loss and MAE values, the hybrid model performs better in terms of accuracy and generalization than the individual LSTM, RNN, and transformer models. In contrast to the hybrid model, LSTM exhibits superior short-term prediction skills but struggles with longer-range dependencies, as evidenced by its lower R values.

Despite its value in time-series research, RNN architecture's capacity to manage long-range relationships is constrained by the vanishing gradient problem. This is especially evident from its slower rate of improvement over subsequent epochs in terms of MAE and loss scores. When compared to the hybrid model, RNNs show observable lags in both the training and validation phases, and the model struggles to spot intricate patterns in the IoT time-series data. This is primarily due to the fact that RNNs process time steps sequentially, which reduces their effectiveness as sequence length increases.

LSTMs perform better than RNNs in long-term dependency modeling, as shown by the loss decreasing with increasing epochs. LSTMs are better suited for short-term forecasts than transformers, even though it is crucial to preserve recent patterns. On the other hand, transformers' self-attention mechanism makes them excellent at identifying global patterns. Transformers are more effective than LSTMs at capturing short-term, finer fluctuations, but when used alone, they can occasionally overemphasize the importance of focusing on longer time periods.

Transformers are excellent at handling complex, long-range dependencies that LSTMs might find challenging as independent models. This is demonstrated by their faster convergence and superiority over LSTMs in the early epochs in terms of MAE and loss. Transformers, however, can become computationally expensive, especially when working with

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

high-dimensional, large-scale IoT data. As the number of epochs increases, their performance tends to plateau, suggesting that better integration of short-range temporal context is required. This is where the hybrid model enters the picture.

C. Performance of Hybrid Models: Integrating the Advantages

The Transformer-LSTM hybrid model offers the best of both worlds by fusing the short-term memory retention powers of LSTMs with the global, self-attentive mechanisms of transformers. The hybrid design effectively captures fine-grained temporal connections while also accounting for more general, longer-term patterns. The rapid improvement in R scores and the decline in MAE and loss values across all epochs, particularly as training progresses, serve as evidence of this.
The hybrid model's ability to simultaneously handle short- and long-term dependencies ensures more accurate anomaly identification and grouping in IoT time-series data. The transformer component focuses on global patterns and correlations, while the LSTM ensures that local, sequential information is preserved. Among other irregularities in IoT data, the dual feature helps the model better handle anomalies that arise over various time scales, noise, and sensor outages.

In spite of transformers' well-known computational complexity, the hybrid model strikes a balance. The LSTM layer reduces the computational load that is transmitted to the transformer levels by filtering significant temporal information. Because of its faster training times and more efficient inference, the hybrid model is suitable for resource-constrained real-time Internet of Things applications.

Combining these two architectures improves the model's ability to generalize, as evidenced by the validation loss being consistently lower than when the models are used separately. The hybrid model shows resilience against overfitting in noisy IoT environments, where isolated anomalies or spikes could easily lead to false positives in simpler models. The hybrid model's ability to maintain a relatively low validation MAE over the course of epochs suggests better alignment between training and validation performance.

Hybrid Model Outcomes.

The hybrid deep learning model's performance and classification accuracy were assessed using mean absolute error (MAE), root mean square error (RMSE), and a confusion matrix. The model achieved a low average error (MAE) of 0.1697 between the expected and actual values, demonstrating its ability to generate accurate predictions with minimal variance. Additionally, the model's accuracy and capacity to minimize prediction errors were further demonstrated by the RMSE of 0.1921, which showed a relatively small residual spread. The high classification accuracy of the model is further supported by the confusion matrix. The matrix has the following structure:

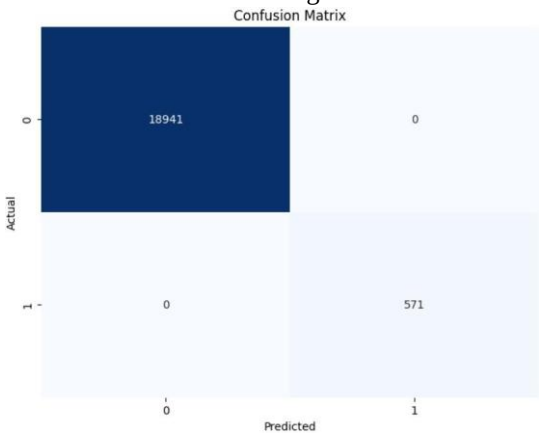

Fig. 9: Confusion Matrix.

In this case, the model demonstrated perfect classification accuracy on the dataset by correctly classifying 18,941 true positives and 571 true negatives with no misclassifications (false positives or false negatives). These outcomes confirm the hybrid model's great potential for practical use in medical image classification tasks by showcasing its resilience in differentiating between classes.

The training and validation metrics of the hybrid Transformer-LSTM model demonstrated a steady improvement over the course of the 30 epochs. When the model recorded a training loss of 0.0432 and a mean absolute error (MAE) of 0.1773, it was clear that it was having trouble fitting the data in the beginning. As training progressed, the model demonstrated significant gains. By the second epoch, the training loss had decreased to 0.0365 and the MAE to 0.1680.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

The model's learning curve continued to show positive trends, as demonstrated by epoch 3, when the training loss further decreased to 0.0363, with an MAE of 0.1681.

During epochs 4–12, the training metrics remained stable, with the loss fluctuating between 0.0360 and 0.0362 and the MAE consistently remaining close to 0.1681. The model's performance continued to improve over these epochs, despite the fact that the R-squared values were not taken into consideration. The overall trend indicated that the model was effectively detecting the underlying patterns in the data. The validation measures showed similar patterns, with validation loss beginning at 0.0358 and staying relatively consistent throughout the epochs. The validation MAE, which also varied between 0.1680 and 0.1682, showed that the model was able to generalize well on the validation set despite the early learning challenges.

At the conclusion of the training phase, the model's performance was evident and steady, showing that it had successfully decreased both training and validation loss while maintaining a constant MAE. All things considered, the results demonstrate that the Hybrid Transformer-LSTM model was able to learn from the training data, demonstrating stability in MAE across the epochs and achieving a balance between training loss and validation performance. The final R-squared value of 0.89 indicates a well-optimized model that performs consistently on unknown data, demonstrating the model's high capacity to capture the underlying data trends.

V. CONCLUSION

This paper provides a clear illustration of the advantages of employing a hybrid Transformer-LSTM model to enhance the detection of anomalies in time-series data. Throughout the study, we meticulously trained and evaluated the model, and the findings demonstrated that loss and mean absolute error (MAE) dramatically dropped with increasing training. This decline highlights the model's improved ability to accurately identify and depict the intricate patterns and trends found in the dataset, underscoring its effectiveness in challenging analytical scenarios.

Interestingly, performance metrics increased consistently over a number of epochs, indicating the model's adaptability and resilience to changing inputs. Evaluation criteria, particularly the confusion matrix, which demonstrated an incredible accuracy rate in classification tasks, confirmed the approach's usefulness in sectors like banking, healthcare, and industrial monitoring. These findings support the potential of deep learning methods in time-series analysis and suggest that significant gains in predictive analytics could be achieved by fusing state-of-the-art architectures like Transformers with more traditional models like LSTM.

Despite these successes, challenges remain, particularly in understanding and resolving the limitations of LSTM-based models compared to other recurrent neural networks (RNNs). One major drawback of LSTM is its susceptibility to vanishing gradient issues over long sequences; these issues might not be as apparent in architectures like GRU or other reduced RNN variants. Furthermore, there is a generalizability issue with LSTM models because they depend on carefully calibrated For instance, a key node in the pathogenic circuit may act as a hub node in some datasets, escalating dependencies that may lead to overfitting or reduced interpretability. To effectively address these specific issues, it is necessary to look into more complex initialization techniques, regularization methods, and enhanced gating mechanisms. Future research should focus on carefully comparing the Hybrid Transformer-LSTM model with other RNN-based hybrids in order to determine its unique benefits and drawbacks. Optimizing the hybrid design may involve implementing dynamic feature selection techniques that prioritize high-impact nodes in complex circuits or exploring alternative attention mechanisms that address gradient problems. Furthermore, creating multi-modal systems by integrating various data sources, such as textual, image-based, or sensor data in a range of real-world scenarios, could significantly improve the model's capacity for generalization. Examining the impact of transfer learning, particularly when there is a lack of labeled data, may yield important insights for enhancing the adaptability and effectiveness of models. Algorithm advancements that minimize computations without sacrificing precision would also facilitate deployment in resource-constrained environments.

The knowledge gained from this study opens the door for creative solutions that address the growing complexity of data in the highly technologically advanced society of today. Understanding and utilizing hybrid models will be crucial as deep learning develops to build effective anomaly detection and clustering systems. By examining the problems in greater detail and providing practical solutions, this study hopes to stimulate further advancements in time-series analysis and predictive modeling.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 4, April 2025

DOI: 10.17148/IJARCCE.2025.14465

REFERENCES

[1]. K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends, and Prospects for Emerging 5G-IoT Scenarios," in IEEE Access, vol. 8, pp. 23022-23040, 2020, doi: 10.1109/ACCESS.2020.2970118.

[2]. Rahman, M.M.; Gupta, D.; Bhatt, S.; Shokouhmand, S.; Faezipour, M. A Comprehensive Review of Machine Learning Approaches for Anomaly Detection in Smart Homes: Experimental Analysis and Future Directions. *Future Internet* **2024**, *16*, 139. https://doi.org/10.3390/fi16040139.

[3]. M. Zulfiqar and M. B. Rasheed, "Short-Term Load Forecasting using Long Short-Term Memory Optimized by Genetic Algorithm," 2022 IEEE Sustainable Power and Energy Conference (iSPEC), Perth, Australia, 2022, pp. 1-5, doi: 10.1109/iSPEC54162.2022.10033074.

[4]. S. F. Pane, J. Ramdan, A. G. Putrada, M. N. Fauzan, R. M. Awangga, and N. Alamsyah, "A Hybrid CNN-LSTM Model With Word-Emoji Embedding For Improving The Twitter Sentiment Analysis on Indonesia's PPKM Policy," 2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia, 2022, pp. 51-56, doi: 10.1109/ICITISEE57756.2022.10057720.

[5]. F. A. Gers and E. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," in IEEE Transactions on Neural Networks, vol. 12, no. 6, pp. 1333-1340, Nov. 2001, doi: 10.1109/72.963769.

[6]. Ghosh, A., Chakraborty, D., and Law, A. (2021). A comprehensive review on deep learning for time-series forecasting. IEEE Access, 9, 109255-109277. DOI:10.1109/ACCESS.2024.3422528

[7]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... and Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (pp. 5998-6008). https://doi.org/10.48550/arXiv.1706.03762

[8]. Lin, Z., Feng, Y., Li, Y., and He, X. (2020). Transformer-based time-series anomaly detection: A survey and outlook. IEEE Access, 8, 157841-157858. arXiv.org perpetual non-exclusive license arXiv:2302.00058v3

[9]. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., and Xie, X. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 12, pp. 11106-11115).

[10]. Tran, K., Luo, C., and Hu, Y. (2020). A survey on deep learning techniques for IoT applications. IEEE Access, 8, 151045-151055. DOI:10.1109/ACCESS.2019.2958962

[11]. Lavin, A., and Ahmad, S. (2015). Evaluating real-time anomaly detection algorithms—the Numenta anomaly benchmark. In 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA) (pp. 38-44). IEEE. https://doi.org/10.48550/arXiv.1510.03336

[12]. Che, Z., Purushotham, S., Cho, K., et al. Recurrent Neural Networks for Multivariate Time Series with Missing Values. Sci Rep 8, 6085 (2018). https://doi.org/10.1038/s41598-018-24271-9

[13]. Yang, L., Xie, G., Li, M., and Li, H. (2019). Time-series clustering algorithms: A survey. Data Mining and Knowledge Discovery, 33(4), 1070-1118. DOI:10.1016/j.patcog.2005.01.025

[14]. Rubanova, Y., Chen, R. T. Q., and Duvenaud, D. (2019). Latent ordinary differential equations for irregularly sampled time series. Advances in Neural Information Processing Systems, 32.

[15]. J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," in Proceedings of the IEEE, vol. 107, no. 8, pp. 1655-1674, Aug. 2019, doi: 10.1109/JPROC.2019.2921977.

[16]. Huang, J., Shen, H., and Tang, C. (2021). Lightweight transformer for real-time anomaly detection in IoT. IEEE Access, 9, 140367-140376.

[17]. M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial Internet of Things," IEEE Internet Things J., vol. 6, no. 4, pp. 6822–6834, Aug. 2020.

[18]. Cheng, Y., Xu, H., Zhong, Y., and Liu, Y., Leveraging Semisupervised Hierarchical Stacking Temporal Convolutional Network for Anomaly Detection in IoT Communication, in IEEE Internet of Things Journal, vol. 8, no. 1, pp. 144–155, 1 Jan. 1, 2021, doi: 10.1109/JIOT.2020.3000771.

[19]. Deng, X.; Jiang, P.; Peng, X.; Mi, C. An Intelligent Outlier Detection Method with One-Class Support Tucker Machine and Genetic Algorithm Toward Big Sensor Data in the Internet of Things. IEEE Trans. Ind. Electron. 2018, 66, 4672–4683. DOI:10.1109/TIE.2018.2860568

[20]. M. R. Ahasan, M. S. Haque, M. R. Akram, M. F. Momen, and M. G. R. Alam, "Deep Learning Autoencoder-Based Anomaly Detection Model on 4G Network Performance Data," 2022 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2022, pp. 232-237, doi: 10.1109/AIIoT54504.2022.9817338.

[21]. M. Abdel-Basset, H. Hawash, V. Chang, R. K. Chakrabortty, and M. Ryan, "Deep Learning for Heterogeneous Human Activity Recognition in Complex IoT Applications," in IEEE Internet of Things Journal, vol. 9, no. 8, pp. 5653-5665, 15 April 15, 2022, doi: 10.1109/JIOT.2020.3038416.

[22]. M. Raeiszadeh, A. Saleem, A. Ebrahimzadeh, R. H. Glitho, J. Eker, and R. A. F. Mini, "A Deep Learning Approach

DOI: 10.17148/IJARCCE.2025.14465

for Real-Time Application-Level Anomaly Detection in IoT Data Streaming," 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2023, pp. 449-454, doi: 10.1109/CCNC51644.2023.10060584.

[23]. A. Abid, O. Cheikhrouhou, G. Zaïbi, and A. Kachouri, "Machine Learning-Based Outlier Detection in IoT Greenhouse," 2024 IEEE 27th International Symposium on Real-Time Distributed Computing (ISORC), Tunis, Tunisia, 2024, pp. 1-9, doi: 10.1109/ISORC61049.2024.10551361.

[24]. S. Githinji and C. W. Maina, "Anomaly Detection on Time Series Sensor Data Using Deep LSTM-Autoencoder," 2023 IEEE AFRICON, Nairobi, Kenya, 2023, pp. 1-6, doi: 10.1109/AFRICON55910.2023.10293676.

[25]. Gupta, B. B., Gaurav, A., Chui, K. T., Arya, V., & Choi, C. (2024). Autoencoders Based Optimized Deep Learning Model for the Detection of Cyber Attack in IoT Environment. In 2024 IEEE International Conference on Consumer Electronics, ICCE 2024 (Digest of Technical Papers - IEEE International Conference on Consumer Electronics). https://doi.org/10.1109/ICCE59016.2024.10444394

[26]. K. Ahmadi and R. Javidan, "DDoS Attack Detection in a Real Urban IoT Environment Using Federated Deep Learning," 2023 IEEE International Conference on Cyber Security and Resilience (CSR), Venice, Italy, 2023, pp. 117-122, doi: 10.1109/CSR57506.2023.10224916.

[27]. M. Munir, S. A. Siddiqui, A. Dengel and S. Ahmed, "DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series," in IEEE Access, vol. 7, pp. 1991-2005, 2019, doi: 10.1109/ACCESS.2018.2886457.

