Innovative Computing Review (ICR)
Volume 6 Issue 1, Spring 2026
ISSN$_{(P)}$: 2791-0024, ISSN$_{(E)}$: 2791-0032
Homepage: https://journals.umt.edu.pk/index.php/ICR

Article QR

Title:	**Explainable Deep Learning Approach: Interpretation Between Detection and Root Cause Analysis of Distributed Log Anomalies with TLAN-EX**
Author (s):	Iman Nasir*, Maryam Tahir, and Seemab Firdous
Affiliation (s):	Department of Information Technology, Faculty of Computing, University of Gujrat, Pakistan
DOI:	https://doi.org/10.32350/icr.61.01
History:	Received: January 19, 2026, Revised: February 23, 2026, Accepted: April 20, 2026, Published: June 25, 2026
Citation:	I. Nasir*, M. Tahir, S. Firdous, "Explainable Deep Learning Approach: Interpretation Between Detection and Root Cause Analysis of Distributed Log Anomalies with TLAN-EX," *Innov. Comput. Rev.*, vol. 6, no. 1, pp. 01-13, December. 2026, doi: https://doi.org/10.32350/icr.61.01.
Copyright:	© The Authors
Licensing:	This article is open access and is distributed under the terms of Creative Commons Attribution 4.0 International License
Conflict of Interest:	Author(s) declared no conflict of interest

A publication of
School of Systems and Technology
University of Management and Technology, Lahore, Pakistan

Explainable Deep Learning Approach: Interpretation Between Detection and Root Cause Analysis of Distributed Log Anomalies with TLAN-EX

Iman Nasir*, Maryam Tahir, and Seemab Firdous

Department of Information Technology, Faculty of Computing, University of Gujrat, Pakistan

ABSTRACT The deeper analysis into modern distributed banking environments has discovered that nowadays distributed log is facing challenges. This is because transaction data comes from multiple networked nodes, which makes a complete system behavior check difficult. This shows how complications grow when multiple system components produce logs at the same time. This gives rise to data inconsistencies that cause anomaly detection to be more difficult. The detection systems used nowadays show lower reliability due to high log volumes and logs that do not follow the standard format, and events showing no clear order, due to which the system raises alarms and fails to catch the real anomalies. Research into log anomaly detection by using Deep Learning (DL) models has shown positive results. However, distributed systems make it hard and difficult for real-life implementation. The current sequence models used in anomaly detection systems only identify one pattern type at a time between short-term and long-term patterns. This results in lower accuracy when different nodes produce dependent log data. The existing detection methods fail to give post-hoc explanations. This makes it hard for the analysts to understand the causes of anomalies, their starting point, and the cause. The current research is based on previous researches. The study implemented the TCN-Transformer hybrid system, which would be able to find patterns at both short and long-time scales in distributed log data. The system employs Parameter Entity Labeling (PEL) to enhance the working and quality of the preprocessing operations of the logs. The model adds three new elements, which include a post-hoc attribution module through SHAP and Integrated Gradients, a propagation analysis layer, and an overhead evaluation module. The model would result in improving the transparency and performance and might be able to cope with the issues as they arise. The designed (TLAN-EX) Temporal-Logical Attention Network with Explainability model has the capability to achieve high accuracy and present clear explanations and low response time. The study revealed that the hybrid learning approach assists in enhancing multi-node anomaly detection and SHAP and Integrated Gradients help to justify the causes of anomalies. The propagation analysis is used to demonstrate the spread of system anomalies between nodes to indicate the detailed working pattern of the system. The system is analyzed to be at its optimal level as the model is highly functioning when the system is considered to be handling high traffic. The upgraded explanation capabilities and speed of detection of the model depict improved detection results and suit well in distributed financial systems.

INDEX TERMS anomaly detection, deep learning, distributed system logs, explainable AI, post-hoc attribution, TCN-transformer, temporal logic model

*Corresponding Author: 24025956-006@uog.edu.pk

I. INTRODUCTION

The current distributed systems deployed in the financial sector and in large-scale infrastructures rely on ongoing log analysis to ensure good performance [1]. The digital footprint of application behavior resides in high-volume, time-stamped logs that serve as the primary system event documentation. The combination of complex microservice and networked systems produces enormous log data volumes with unorganized structures and distributed event relationships that make human-based anomaly detection impossible and require automated solutions.

Deep Learning (DL) methods used for log anomaly detection [2] encounter two essential performance limitations. The current DL methods fail to detect all temporal relationships. This is because they concentrate on either brief execution sequences or complete system-wide patterns. The system generates logs from complex multi-node systems, which results in reduced detection accuracy due to this limitation. Most systems function as untraceable black-box systems. This prevents security analysts from using post-hoc explainability to identify anomaly origins, perform system audits, and implement defensive measures.

The current study presented a temporal log attention network with explainability TLAN-EX as a new hybrid system that solves both temporal dependency modeling and explainability problems in log anomaly detection. The TLAN-EX model uses Temporal Convolutional Network (TCN) and transformer encoder components to detect both short-term execution patterns and extended system-wide trends in distributed log data. The system provides transparent explanations for all predictions through its mandatory post-hoc attribution functionality.

A. HYBRID TEMPORAL-LOGICAL ARCHITECTURE

The TLAN-EX combines TCN dilated convolutions for effective local feature extraction with transformer self-attention for long-range dependency modeling, improving log anomaly detection across diverse anomaly types.

The Parameter Entity Labeling (PEL) strategy enhances the preprocessing of logs in transforming Universally Unique Identifier (UUID) and timestamps into meaningful tokens. This would assist in the filtering of unnecessary vocabulary noise. Moreover, it would also enable the systems to learn log templates in a more efficient manner.

New post-hoc attribution techniques are also suggested in the proposed model, and provide scores on the importance of features. Using these scores assists the analysts in knowing the place of origin of the anomaly and the parameter that causes this anomaly.

TLAN-EX is tested against benchmark log datasets. The findings indicate that it has a superior performance compared to most of the current state-of-the-art approaches.

The study is presented as follows:. Section 2 includes a review of related literature on log parsing, techniques of anomaly detection, and model explainability based on DL. In section 3, the TLAN-EX model architecture is demonstrated as to how it works as well as its implementation steps. The results and findings are discussed in section 5. Section 6 presents a conclusion and outlines the potential future research directions.

II. LITERATURE REVIEW

Anomaly detection in distributed [3] computing has become an important field of research due to highly complicated log systems. Logs, which record events and conditions of distributed systems, are considered as a key resource for pointing out abnormal patterns that show security threats [4]. When the older and traditional ways of finding log problems are compared, such as simple grouping or statistics, they do not work so well when the system is big, and everything is connected. To resolve this problem, other researchers have started using Natural Language Processing (NLP) with DL. This is a smarter way to pull out important data automatically and find problems more accurately.

In the study [5], authors created a hybrid Convolutional Neural Network (CNN) and bidirectional long short-term memory (BiLSTM) network to enhance network intrusion detection using binary and multiclass classification. On the side of detection, their model performed very well. However, it failed entirely on the interpretability component. It was simply a Black Box, and the very cause of its detection could not be identified.

In the study [6], authors concentrated on the resolution of the cleaning log data of a complex type. They applied a CNN model with PEL, which does not rearrange the logs to a loss of their meaning. The fact that CNN only examined the fixed elements of the log sequence was the only issue. As the behavior of the system varied, the model became trapped and was unable to accommodate the new changes very effectively.

In reference [7], researchers attempted to make it better by incorporating models of big size, such as Roberta and Long former within their model. Their models, however, were so heavy on the system. They did not provide the real cause of the issue. Operating such a structure in a distributed environment is simply too risky or slow.

In the study [8], the researchers also considered NLP, they simply paid attention to making the log messages readable in a more decent manner. Furthermore, they did not even note the time factor. They did not completely demonstrate how to trace abnormalities that cross over multiple servers. Their system was also short-changed since it lacked a clear explanation on the process of tracing the anomaly when it propagated to the entire distributed network [1].

Another important study was published in [9]. The authors in the respective study came up with a hybrid model using transformer and TCNs. This model was excellent at capturing all the timing relationships in the logs. However, their model had a big problem: it only gave a very limited explanation about how to track the anomaly when it spreads across the whole distributed system.

In the study [10], hybrid DL approaches have been mentioned, such as CNN-BiLSTM, for network intrusion detection.

Recent work [2] has used Machine Learning (ML) and DL methods of Distributed Denial of Service (DDoS) anomaly detection in Software Defined Networking (SDNs).

All these studies indicate that DL is certainly the solution to the problems in detecting log anomalies. However, these studies also indicate the drawbacks in explaining things, finding cause and effect step by step, and getting the model to run swiftly enough. TLAN-EX has been suggested in order to address these

constraints. TLAN-EX would provide all the missing links in a single DL framework. This model takes into consideration the meaning of the log text, the timing relations, and the detailed explanation. This model would make detection more accurate and increase clarity in the field of banking.

A. PROPOSED MODEL

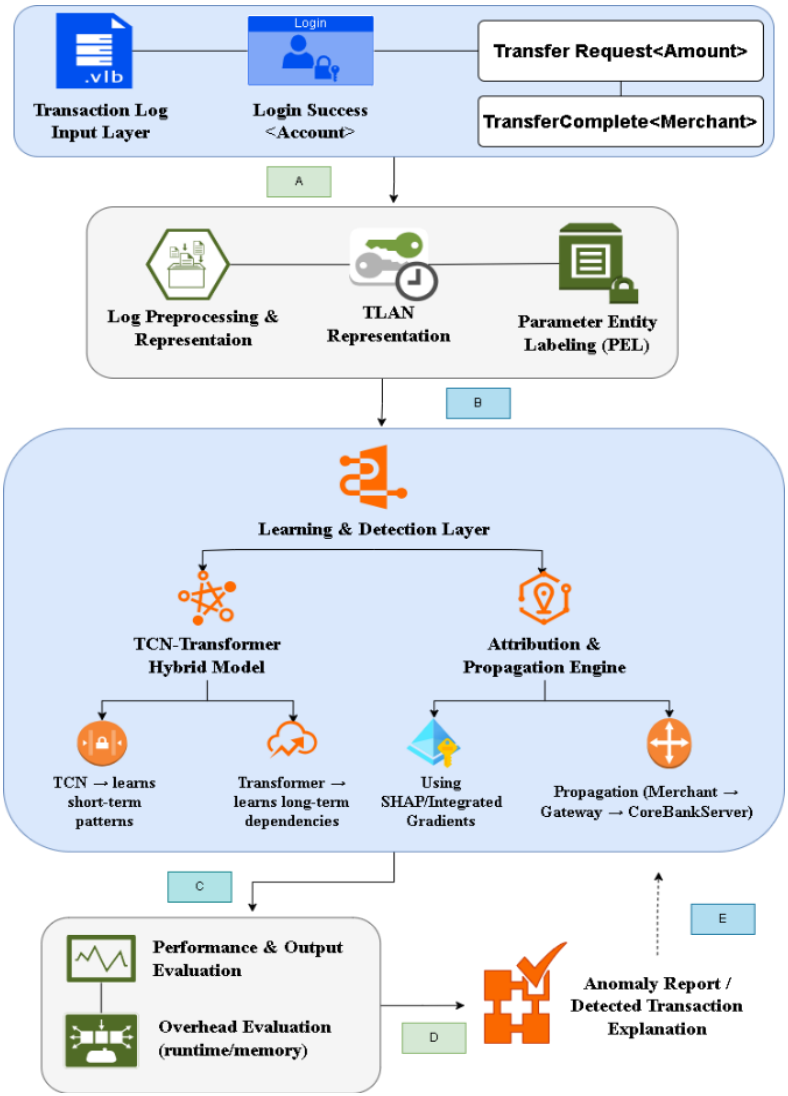

FIGURE 1. Proposed model TLAN-EX architecture to find Anomalies in the distributed banking transaction system. (A) represents the input layer of a transaction log, (B) then log. Preprocessing input layer, where the log is transformed into a TLAN, (C) the learning and detection layer, where a hybrid TCN-Transformer is used, (D) the post-hoc attribution and propagation analysis, and (E) shows the performance and overhead analysis.

III. METHODOLOGY

The proposed TLAN-EX model was planned as a hybrid DL design to identify issues in distributed log systems and to describe them. The entire system was found on the cleaning up of the raw logs, learning the patterns, detecting the irregularities, and once detected, the model provides a clear picture in a graph format. As shown in Fig.1, each module was assigned a particular task to perform in the overall process, which makes the model robust, efficient, and brings out anomalies.

The process started with (A) Transaction Log Input Layer processing and normalization of logs, as shown in Fig.1, where the system gathered logs in various locations. Such logs consisted of event IDs, time stamps, transactions, and system responses. These logs had to be prepared to be analyzed. The PEL method (B) was employed to transform attributes, such as account numbers, transaction IDs, and user IDs into label patterns rather than sensitive information. It was also the best way of minimizing additional noise, equalizing the data format, and improving compatibility between the system and various nodes.

After processing, (B) the semantic embedding layer converted the labeled log templates to numbers. This step used deep embedding techniques [11] in order to obtain structural meaning of the log results.

Embedding used showed the importance of the connections which were kept with the important events of the log (B) and structural meanings of log events (A). These are then sent to TLAN where more sophisticated features are acquired.

The important aspect of the system was the TLAN module (C), which united a TCN and transformer encoder to enable the model to acquire various degrees of dependencies. The TCN component was used to detect anomalies, abrupt changes, or repeat transactions, whereas the transformer identified patterns and trends in the entire system over time. This study balanced both elements in an attempt to ensure that TLAN-EX aligns itself between concentrating on local specifics and seeing the overall picture. This provides a complete picture of how log behaviors vary over time.

Once the system identified an anomaly, the model proceeded to the post-hoc attribution and Propagation phase (E), which was used to make the model easy to understand. It relied on approaches, such as SHAPLEY, Additive explanations, and SHAP [12], and Integrated Gradients [13]. The anomalies were attributed to factors such as abnormal transaction amounts and unauthorized access by multiple nodes.

The result of the attribution module got into the propagation explanation module (E). This module described the spreading of the anomaly detected into various parts of the system. This helped the system to realize how one unnatural event may become a bigger issue for the entire system and provide effective safety precautions.

Finally, the effectiveness of the model was investigated by the top-layer of the overhead evaluation (D). It quantified variables, such as processing time, memory usage as well as computing costs to guarantee that the add-on of explainability attributes did not make things slower. On one hand, this layer provided balance and thus made the model transparent and on the other hand, the system was able to handle more work.

TLAN-EX was appropriate in real-time application. The findings indicated that the model was not sluggish in processing whilst being speedy and clear.

IV. RESULTS

A. LOG PREPROCESSING AND NORMALIZATION

The initial process involved logging, preprocessing, and normalization, where all the incoming logs were taken in the same format. PEL was employed as the primary method of transformation in this phase. A special tokenizer was employed to locate variable parts, such as user IDs, transaction references, time stamps or node IDs, and substituted them by the labels, such as accounts or transactions. This change was achieved by using Python-based regular expressions and text-parsing tools. The logs after normalization were then stored in a lightweight distributed storage system for the next module processing.

B. SEMANTIC EMBEDDING

The semantic embedding converts the labeled textual logs into strong vector representations. Every split event gets turned into a vector through models that understand context, such as Word2Vec or transformers. Instead of simple tools, frameworks, such as PyTorch or TensorFlow manage this mapping to capture word patterns along with surrounding meaning in logs. The embedding functions would be the learning feature for the next modules. Moreover, these functions would ensure that the modules capture both the meaning and the temporal context of each and every transaction.

The embedded sequence $\boldsymbol{S}$ of length m Is transformed into.

The input matrix $\boldsymbol{E}$

$$\boldsymbol{E} = \text{Embedding}(\boldsymbol{S}) = [\boldsymbol{e_1}, \boldsymbol{e_2}, \dots, \boldsymbol{e_m}] \quad (1)$$

The embedding dimension D_{emb} was set to 128 **(**Table I**).** This ensured that the vectors captured sufficient information for the downstream sequence modeling modules [6].

TABLE I

LOG PREPROCESSING MODULE: PARAMETER ENTITY LABELING (PEL)

Component	Parameter	Description	Default Value
Log Windowing	Sequence Length (L)	The maximum number of consecutive log events is considered as one input sequence for the model.	128
PEL	Numeric Pattern	Regex to replace all sequences of digits with a token.	\\d+ (Replaced with <NUM>)
	Alphanumeric ID Pattern	Regex to replace long, mixed-case strings with a token.	[A-Za-z0-9]{8,} (Replaced with <ID>)
	Time/Date Pattern	Regex to mask timestamps and date formats common in	[0-9]{4}-[0-9]{2}-[0-9]{2}\\s[0-9]{2}:[0-9]{2}:[0-9]{2}

Component	Parameter	Description	Default Value
		banking/system logs.	(Replaced with <TIME>)
Embedding	Embedding Dimension (D_{emb})	The size of the vector representation for each log token (post-PEL).	128
	Embedding Initialization	The pre-trained word embedding is used to initialize token vectors.	Word2 Vec (trained on corpus)

C. TEMPORAL-LOGICAL ATTENTION NETWORK (TLAN)

The TLAN is implemented as the core computational engine of the model. It is built using a hybrid neural network design which includes TCNs and transformer encoders within the same architecture. The TCN layer is used to find local time-based patterns, and it works by applying 1D operations with causal padding to make sure future events do not affect past calculations. This design helps the network focus on short-term dependencies. On the other side, the transformer encoder uses self-attention to capture long-term patterns and relationships between events in the logs. The outputs from both parts are combined, then passed through normalization and dropout layers to prevent overfitting. After that, they go to a fully connected classification layer to prevent overfitting. After that, they go to a fully connected classification layer to predict anomalies.

D. TCN FOR LOCAL FEATURE EXTRACTION

The TCN part is responsible for finding local time-based patterns and short-term dependencies. It works using 1D convolutional operations with causal padding, making sure that future events do not affect the calculation of past events [9]. The output y_t of a dilated causal convolutional at time t is calculated as:

$$\boldsymbol{y_t = \sum_{k=0}^{K-1} w_k \cdot x_{t-d \cdot k}} \quad (2)$$

The configuration settings for the TCN module, which use a residual block structure [9], are given in (Table II).

TABLE II

TEMPORAL CONVOLUTIONAL NETWORK (TCN)

Parameter	Description	Default Value
Number of TCN Layers	The number of stacked residual blocks.	4
Kernel Size (K)	The size of the 1D causal convolutional filter.	3
Dilation Rates ($\text {DR}$)	The dilation factors increase exponentially to expand the receptive field.	$[1, 2, 4, 8] $

Parameter	Description	Default Value
Filter Count (C)	The number of output channels (filters) per TCN layer.	64
Dropout Rate ($\text {TCN}_{\text{dropout}}$)	Dropout is applied within the residual blocks.	0.1

E. TRANSFORMER FOR GLOBAL DEPENDENCY MODELING

The output of the TCN layer is passed to the transformer.

Encoder block captures long-term dependencies and logical event relationships across the distributed logs using the multi-head self-attention mechanism.

The core scaled dot-product attention function [14] is defined as:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\boldsymbol{T}}}{\sqrt{\boldsymbol{d_k}}}\right)\boldsymbol{V} \quad (3)$$

In (Table III), the configuration of the transformer encoder is specified as:

TABLE III

TRANSFORMER ENCODER

Parameter	Description	Default Value
Number of Encoder Layers (N_{enc})	The number of identical stacked Transformer Encoder blocks.	2
Number of Attention Heads (H)	The number of parallel attention mechanisms in the Multi-Head Attention layer.	8
Hidden Dimension (D_{model})	The dimensionality of the input and output (matches D_{emb}).	128
Feed-Forward Dimension (D_{ff})	The size of the inner layer in the point-wise feed-forward network.	256
Dropout Rate ($\text{Trans}_{\text{dropout}}$)	Dropout is applied to attention weights and layer outputs.	0.1

F. ANOMALY SCORING VIA RECONSTRUCTION

During training, the model is trained to minimize the reconstruction loss. $\mathcal{L}Recon(\theta)$ [5], [9]

$$\mathcal{L}\text{Recon}(\boldsymbol{\theta}) = \frac{\mathbf{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}}|\boldsymbol{e_i} - \hat{\boldsymbol{e}}_{\boldsymbol{i}}|^{\mathbf{2}} \quad \ldots(4)$$

During inference, an anomaly score A is calculated. A log sequence $\boldsymbol{S}$ is flagged as anomalous if its reconstruction error exceeds a predetermined threshold τ:

$$A(S) = \frac{1}{m}\sum_{i=1}^{m}|e_i - \hat{e}_i|^2 > \quad \ldots(5)$$

G. POST-HOC ATTRIBUTION MODULE

Once the model predicted an anomaly, the post-hoc attribution module was activated to explain the main cause of the detection. This module operated independently of the model's accessible methods, such as SHAP and Integrated Gradients. In a banking system, the module might find that the amount of the merchant was the main reason for triggering a fraud alert. This explanation process was done using open-source tools for explainability, which were added to the neural network framework. The explanations were stored in an organized way, so that they can be easily reviewed by analysts and used for system audits.

The SHAP value ϕ_i for feature i was calculated as:

$$\phi_i(f,x) = \sum_{S\subseteq x\setminus\{i\}} \frac{|S|!\,(|x|-|S|-1)!}{|x|!}[f(S \cup \{i\}) - f(S)] \quad (6)$$

This process identified specific log templates or parameters, such as an unusual transaction amount or repeated access from a certain node that had the biggest impact on the anomaly score [12], [13] (Table IV) gives a clear measure of how important each feature is.

TABLE IV

POST-HOC ATTRIBUTION MODULE (SHAP/INTEGRATED GRADIENTS)

Component	Parameter	Description	Default Value
Integrated Gradients	Number of Steps	The number of samples used along the path from the baseline to the input to approximate the integral.	50
	Baseline Input	The reference input against which attribution is measured.	Zero-vector embedding
SHAP	Number of Samples	The number of background samples used for the Monte Carlo approximation of Shapley values.	500
	Background Dataset	A representative subset of normal log sequences is used to define the "average" model output.	10% of Normal Training Data

H. PROPAGATION EXPLANATION MODULE

Once the propagation explanation module was assigned to the attribution, it became possible to know the impact of the anomaly on the entire network. This was graph-based, which indicates how an anomaly in one area of the system disseminates to other related areas. The system constructed an outgoing graph without any cycles by following the dependency between events, using transaction records and system

messages. Nodes were parts or actions in the system, whereas the relationships between them indicated cause and effect. The process also helped in determining the root cause of the problem and also helped in automatic system defense by proving the source and spreading of the aberration.

I. OVERHEAD EVALUATION

The last activity, overviewed evaluation, made sure that the model could be applied in the real-life context because it satisfies the criteria of the model to make its calculations. Here, memory consumption, processing time, and system throughput of the training and testing were being followed. These facts were collected by using inbuilt-tools in framework of neural network. To maintain the transparency and practicality of the process of identifying the anomalies in a real-world system, the model struck the right balance between transparency and efficiency and took into account such parameters as delay as well as resource consumption.

TLAN-EX has a fluent blend of DL and understandability in the detection of anomalies. Each of the components including the preprocessing and the overhead evaluation are components that rectify the errors that have been made in the past models. The preprocessing stage would serve to make the data more similar, and the embedding and hybrid attention networks would guarantee complex time-dependence and logic correlations. How all these components work together is such that TLAN-EX is not only an effective detection model but also an architecture that fills the accuracy versus explainability gap in securing distributed systems.

V. CONCLUSION AND FUTURE WORK

The current study introduced TLAN-EX, which is a new hybrid system, to detect log anomalies in distributed systems. TLAN-EX fuses TCN to get local patterns efficiently and a transformer encoder to get to know more about long-term dependencies. This research also enhanced the construction of PEL to handle noise better when preprocessing and the key post-hoc attribution module SHAP to provide clear and line-by-line explanations on each identified anomaly. Experiments on benchmark datasets demonstrated that TLAN-EX was more accurate in terms of precision, recall, and F1-score. This indicated that it can capture local and global dependencies and also provide understandable explanations.

The future research would revolve around three key aspects: the application of TLAN-EX in real-time, streaming environments with adaptive learning to adapt to changes in the data, adding the explainability module to support counterfactual explanations, and evaluating the capability of the models to address other sequence-based tasks of anomaly detection, such as [10] Network Intrusion Detection Systems (NIDS) and using Large Language Models (LLMs) for better results, as in [15]. The study used LLaMa-based transformer for log detection.

Author Contribution

Iman Nasir: conceptualization, methodology, software, formal analysis, investigation, visualization, writing – original draft. **Seemab Firdous:** methodology, software, validation, formal analysis, investigation, visualization, writing – review & editing. **Maryam Tahir:** conceptualization, resources, validation, writing – review & editing.

Conflict of Interest

The authors of the manuscript have no financial or non-financial conflict of interest in the subject matter or materials discussed in this manuscript.

Data Availability Statement

Data supporting the findings of this study will be made available by the corresponding author upon request.

Funding Details

No funding has been received for this research.

Generative AI Disclosure Statement

The authors did not use any type of generative artificial intelligence software for this research.

REFERENCES

[1] M. Lyu, H. H. Gharakheili, and V. Sivaraman, "A survey on enterprise network security: Asset behavioral monitoring and distributed attack detection," *IEEE Access*, vol. 12, pp. 89363–89383, 2024, https://doi.org/10.1109/ACCESS.2024.3419068.

[2] N. S. Musa, N. M. Mirza, S. H. Rafique, A. M. Abdallah, and T. Murugan, "Machine learning and deep learning techniques for distributed denial-of-service anomaly detection in software-defined networks—Current research solutions," *IEEE Access*, vol. 12, pp. 17982–18011, 2024, https://doi.org/10.1109/ACCESS.2024.3360868.

[3] A. M. Abdallah, A. Alkaabi, G. Alameri, S. H. Rafique, N. S. Musa, and T. Murugan, "Cloud network anomaly detection using machine and deep learning techniques—Recent research advancements," *IEEE Access*, vol. 12, pp. 56749–56773, 2024, https://doi.org/10.1109/ACCESS.2024.3390844.

[4] A. Dehlaghi-Ghadim, M. H. Moghadam, A. Balador, and H. Hansson, "Anomaly detection dataset for industrial control systems," *IEEE Access*, vol. 11, pp. 107982–107996, 2023, https://doi.org/10.1109/ACCESS.2023.3320928.

[5] P. Han, H. Li, G. Xue, and C. Zhang, "Distributed system anomaly detection using deep learning-based log analysis," *Comput. Intell.*, vol. 39, no. 3, pp. 433–455, Jun. 2023, https://doi.org/10.1111/coin.12573.

[6] T. Sutthipanyo, T. Lamsan, W. Thawornsusin, and W. Susutti, "Log-based anomaly detection using CNN model with parameter entity labeling for improving log preprocessing approach," in *Proc. 2023 IEEE Region 10 Conf. (TENCON)*, Chiang Mai, Thailand, 2023, pp. 914–919, https://doi.org/10.1109/TENCON58879.2023.10322478.

[7] J. Shen, R. Tie, Z. Li, B. Liu, Z. Fan, and J. Lu, "Neural network-based log anomaly detection algorithm for 6G wireless integrated cyber-physical system," *Wireless Pers. Commun.*, Jun. 2024, https://doi.org/10.1007/s11277-024-11218-9.

[8] P. Ryciak, K. Wasielewska, and A. Janicki, "Anomaly detection in log files using selected natural language processing methods," *Appl. Sci.*, vol. 12, no. 10, art. no. 5089, May 2022, https://doi.org/10.3390/app12105089.

[9] N. Liao and Z. Liu, "Log anomaly detection method based on Transformer and temporal convolutional networks," *IEEE Access*, vol. 13, pp. 68547–68560, 2025, https://doi.org/10.1109/ACCESS.2025.3561669.

[10] R. Ben Said, Z. Sabir, and I. Askerzade, "CNN-BiLSTM: A hybrid deep learning approach for network intrusion detection system in software-defined networking with hybrid feature selection," *IEEE Access*, vol. 11, pp.

138732–138747, 2023, https://doi.org/10.1109/ACCESS.2023.3340142.

[11] S. Yan *et al*., "Log-based anomaly detection with Transformers pre-trained on large-scale unlabeled data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Denver, CO, USA, 2024, pp. 2059–2064, https://doi.org/10.1109/ICC51166.2024.10623067.

[12] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon *et al*., Eds. Red Hook, NY, USA: Curran Associates, Inc., 2017, pp. 4765–4774.

[13] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, D. Precup and Y. W. Teh, Eds., Sydney, NSW, Australia, vol. 70, 2017, pp. 3319–3328.

[14] A. Vaswani *et al*., "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon *et al*., Eds. Red Hook, NY, USA: Curran Associates, Inc., 2017, pp. 5998–6008.

[15] Z. Yang and I. G. Harris, "LogLLaMA: Transformer-based log anomaly detection with LLaMA," in *Proc. 2025 Int. Joint Conf. Neural Netw. (IJCNN)*, Rome, Italy, 2025, pp. 1–8, https://doi.org/10.1109/IJCNN64981.2025.11227209.