Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

A Dual Hybrid Approach for Log Anomaly Detection using Deep Learning

Harbi Mahmood Abas[a]* ,Ziyad Tariq Mustafa Al-Ta'i[a] and Shumoos jamal Rashid[b]

[a]*University of Diyala, Diyala, Iraq.*
[b]*University of Diyala Presidency, Electronic Computing Center.*
Corresponding author: scicompms242510@uodiyala.edu.iq ,Ziyad1964tariq@uodiyala.edu.iq, Shumoos@uodiyala.edu.iq.

ARTICLEINFO

Article history:
Received: 11 /02/2026
Rrevised form: 10 /04/2026
Accepted : 12 /04/2026
Available online: 30 /06/2026

Keywords:

Deep Learning,

Log Anomaly detection,

CNNs,

Bi-LSTM.

ABSTRACT

The growing complexity of computer software and the explosive growth in the log data have made the detection of anomalies and the identification of problems in the system from the massive logs a major research field. However, existing log anomaly detection techniques cannot identify context-dependent semantic connections in unstructured logs and do not have explicit decision making processes and therefore they cannot be used in very dynamic systems. In this paper, we propose a dual hybrid technique for the anomaly detection task including the feature representation and the classification stages. During the feature representation stage, the statistical TF-IDF technique is used to extract features and merge them with the semantic representations of SBERT to obtain comprehensive representations with both statistical importance and semantic significance. In the classification stage, the hybrid deep learning strategy is used consisting of Convolutional Neural Networks (CNNs) for local pattern extraction and Bidirectional Long Short-Term Memory (Bi-LSTM) network for dealing with temporal relations in the data. This helps to improve the accuracy and effectiveness of the model in detecting anomalies in system logs. The approach has been trained and tested on the commonly used HDFS and BGL datasets. Experimental results showed that the proposed approach achieved an anomaly detection accuracy of up to 0.9989 and therefore demonstrated the potential and effectiveness of this approach compared with previous methodologies.

MSC..

https://doi.org/ 10.29304/jqcsm.2026.18.22645

1. Introduction

Anomaly detection is a major problem in many industries, such as manufacturing, medicine, cybersecurity, and others [1]. It is the process of spotting patterns within data that are different from what is expected or anticipated. Logs are the major source of methods for anomaly detection in almost all computer systems.

Log files are a significant source of information for monitoring computer systems. Accordingly, most of the log events are created as a result of normal system operations, which include process starting and stopping, virtual

∗Corresponding author

Email addresses: *scicompms242510@uodiyala.edu.iq*

Communicated by 'sub etitor'

machine restart, resource access by a user, etc. Additionally, applications produce logs in response to faults or other adverse system conditions such as process failures, availability problems, and security breaches [2].

Complex systems typically monitor their performance by examining log files, which detail every action executed within the system. The heightened complexity of systems and applications generates additional faults and vulnerabilities, ultimately jeopardizing the integrity of the system. Consequently, detecting anomalies has become more challenging, and current procedures are ineffective. A system capable of efficiently detecting anomalies is necessary to ensure an efficient operational environment.

Many approaches for anomaly detection exist, including log parsing [3], wherein various patterns from logs are retrieved and sent to the model for classification of any abnormal activity. These strategies are constrained by the dataset and are ineffective in real-time applications. Certain techniques do not analyze the logs for patterns [4]. They produce sequences of embedding vectors and retain the keys in a list to test the abnormality of the log. Nonetheless, these strategies are ineffective when confronted with unfamiliar data. Such models may yield favorable outcomes when there is significant similarity during testing. To verify this, it is necessary to present low similarity data during testing to assess the model's efficacy in detecting abnormalities.

Machine learning offers many useful techniques for anomaly detection in log files. Several methods have been proposed previously, including clustering [5], log mining [6], statistical analysis of event parameters , and time series analysis [7], among others [8]. Machine learning strategies typically depend on manual feature extraction and are constrained in their capacity to manage highly dimensional or unorganized data.

On the other hand, deep learning techniques learn from raw data and autonomously discover useful features across numerous processing levels. This capacity to learn directly from data enables deep learning models to expand with extensive datasets and address more complex relations.

Lately, researchers have begun employing deep neural networks for log-based anomaly detection, aiming to replicate deep learning's accomplishments in speech recognition and image that surpass traditional machine learning techniques [9]. However, the intrinsically unstructured nature of system events and their complex correlation is a considerable issue to the pre-processing of data to be ingested by neural networks, as well as to the identification of salient features to identify anomalies. Accordingly, the contribution of this study is not limited to using a hybrid CNN-BiLSTM classifier. Instead, it lies in the unified integration of statistical TF-IDF features and semantic SBERT embeddings. Furthermore, the variety of the deep-learning paradigms available, including recurrent and convolutional models, among others, makes the process of choosing the right model to be applied to the specific problem more than nontrivial and makes the process of matching the model requirements to the characteristics of the input data, both structural and statistical, complicated. To address the limitations mentioned above, the study provides an anomaly -identification framework to system logs, by combining a hybrid of deep-learning algorithms, especially CNNs, with Bi-LSTMs. The first step in the preprocessing pipeline will be to refine raw logs data, and then extract statistical and semantic features which are then used to perform categorization using the proposed hybrid learning model.

2. Literature Review

In this section, the literature review referring to the log-based anomaly detection is investigated, especially focusing on the deep learning methods.

M. Du et al. [10] used an LSTM-based log parser to detect patterns in log data during normal operation automatically. This is very adaptive and flexible in its integration across various execution patterns. The model achieved a total performance with an F1-score of 96 % on HDFS log data. One weakness is that it fails to identify abnormalities when log patterns do not match those depicted in the training sample.

Z. Liu et al. [11] used a self-attention neural network to assess anomalies and make decisions. The technique achieved a detection rate of over 89% on a data set carefully compiled from over fifty working network servers, including a specified target server in the campus network. The benefits of this method are that it augments data to address weaknesses in conventional training databases that limit decision-making. The anomaly scores generated are uninterpretable, and the real findings for abnormal log messages may differ.

W. Meng [12] suggested combining LSTM and Template2vec for detecting sequential and quantitative log abnormalities. The method effectively retrieves hidden semantic information in log templates. This study achieved

an F1 score of 0.96 on the BGL dataset and 0.95 on the HDFS dataset. Although it cannot simultaneously detect both quantitative and sequential anomalies, it is less useful in some scenarios.

C. Zhang et al. [13] proposed a hierarchical semantics framework called the LayerLog for detecting anomalies. LayerLog extracted semantic information from each layer. They created semantic vectors for each word using word embedding, POS, and TF-IDF in the first layer. These vectors were sent to the second layer to construct the Word Vector sequence using an attention-based Bi-LSTM model. The final layer generated the Log Vector sequence using an attention-based Bi-LSTM model again used. The system achieved precision, recall, and F1-scores of 0.996, 0.978, and 0.988 on the HDFS and BGL datasets.

Y. Duan et al. [14] created LogEDL, a loss of evidence function for model training. Transformer encoders, ENN heads, and uncertainties make up this structure. They extracted contextual information from the record sequence and created semantic vectors using the transformer encoder. The ENN head converted semantic vectors into record evidence, and the uncertainty assessed anomaly detection based on uncertainty. This work achieved the precision, recall, and F1 score of 90.06, 92.80, and 91.41, respectively, on the HDFS dataset.

A. Aziz and K. Munir [15] presented supervised-unsupervised anomaly detection . Self-organizing maps (SOMs), BERT encoders, and autoencoders were used. Using the BERT encoder to produce semantic vectors and SOMs for clustering. Pattern recognition uses autoencoders. The proposed method had 93% accuracy and 92% recall on HDFS and BGL datasets.

Although there has been a substantial advancement in deep learning anomaly detection systems and methodologies on system logs, a vast majority of research still uses solely statistical or semantically relevant feature representations, and thus limit their generalization ability to new trends. In addition, the models that are currently available face problems that include stability and scalability. Therefore, a hybrid approach is indispensable to overcome such limitations; in this study, we propose a unified hybrid framework that combines statistical and semantic feature fusion with deep learning–based classification.

3. Methodology

This section delineates our framework for anomaly detection. Figure 1 illustrates that the system comprises multiple integrated steps: hybrid feature extraction, preprocessing, and hybrid deep-learning models. All these components will be elaborated upon in the subsequent subsections.

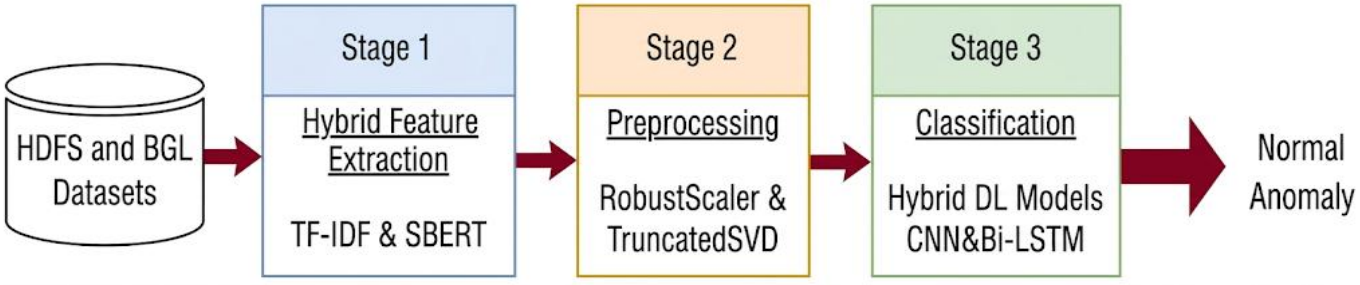


Figure 1: Framework of the proposed model for anomaly detection.
Note that the train/test split is performed before fitting RobustScaler and TruncatedSVD, and only the fitted training transformations are applied to the test data.

3.1. Hybrid Feature Extraction Stage

Data preparation is essential prior to performing feature extraction. Therefore, we eliminated any log files with missing or duplicated data [16]. We used two approaches for extracting statistical and semantic features: ` (TF-IDF)

` to extract statistical features and ` (SBERT encoder) ` to extract semantic features from log files. Then, we hybridized them to improve assessment accuracy and identify anomalies.

Term Frequency-Inverse Document Frequency (TF-IDF): is primarily employed to calculate the similarity of each word's frequency weight, serving as a technique for weighting the target term within that document based on its frequency and significance. The TF-IDF is obtained through the following equations [13]:

$$TF = \frac{|L^i(W_k^{ij})|}{|S_i|} \qquad (1)$$

Where $|L^i(W_k^{ij})|$ denotes the total number of logs in $|S_i|$that encompassW_k^{ij}, and $|S_i|$ denotes the number of logs in sequence.

$$IDF = \lg\left(\frac{|S|}{|S(W_k^{ij})|}\right) \qquad (2)$$

Where $|S|$denote the total number of log sequence in dataset, and $|S(W_k^{ij})|$ denote the total number of S. Then, we compute the TF-IDF value $FreqW_k^{ij}$ of W_k^{ij} using the following formula:

$$FreqW_k^{ij} = \text{TF} \times \text{IDF} \qquad (3)$$

SBERT (Sentence BERT) is an existing pre-trained neural network that is typically used to produce semantic word vectors based on the textual data. Such semantic vectors allow systematic analysis of the log messages, which provides the ability to compare patterns and similarities defining anomalous behavior. The use of Sentence BERT Encoder will improve the detection of anomalies by identifying log message contexts that contain implicit information to be used to detect anomalies compared to traditional feature-based anomaly detection systems [15].

3.2. Preprocessing Stag

This stage enhances data preparedness for anomaly detection. It minimizes computational complexity and directs models toward solely pertinent information. After splitting the dataset into training and testing subsets, TF-IDF features and SBERT embeddings are extracted and concatenated into a unified hybrid feature vector. RobustScaler is then applied to the concatenated hybrid features, followed by TruncatedSVD for dimensionality reduction, using parameters learned from the training set only.

-**Normalization (Robust Scaler):** This normalization scales features using statistics that are robust to outliers. It removes the median and scales the data according to the interquartile range (IQR) by default. In this study, the Robust Scaler is fitted only on the training set, and the learned scaling parameters are then applied to the validation and test sets. The interquartile range (IQR) is the difference between the first quartile (25th percentile) and the third quartile (75th percentile) **[**11**].**

-**Dimensionality Reduction (Truncated SVD):** This estimator performs linear dimensionality reduction by truncated singular value decomposition (SVD). Unlike principal component analysis (PCA), it does not require centered dense data and can work efficiently with sparse matrices [17]. In the proposed pipeline, TruncatedSVD is also fitted only on the transformed training features, and then the same fitted projection is used to transform the validation and test sets. Therefore, no information from the test set is used during normalization or dimensionality reduction, which prevents potential data leakage.

3.3. Classification Stage

The dataset is first partitioned into 70% training and 30% testing subsets. Feature extraction and transformation are then handled in a leakage-safe manner: TF-IDF and SBERT representations are generated, concatenated, and the resulting hybrid feature vectors are normalized and reduced in dimension using parameters learned from the training split only. The fitted transformation pipeline is subsequently applied to the validation and test subsets without re-fitting. After that, the processed features are input into deep learning models to classify normal and anomalous instances in the system logs. We employed the (1D-CNN) and (Bi-LSTM) algorithms and then studied the performance of each model individually and in hybrid form. The models were trained using the Adam optimizer with a learning rate of 1e-3, a batch size of 256, and up to 50 epochs. Because the HDFS and BGL datasets are highly imbalanced, class weights were applied during training to reduce bias toward the majority class. In the implemented

training pipeline, the class weights were computed from the training labels and used during model fitting. No explicit re-sampling technique, such as over-sampling or under-sampling, was applied in this study; instead, the original data distribution was preserved and imbalance was handled through weighted learning [18][19].

To further improve transparency and reproducibility, the overall experimental workflow of the proposed framework is summarized as follows:

Experimental Workflow of the Proposed Pipeline:

1. Load and clean the raw log dataset by removing missing or duplicated records.
2. Split the dataset into training (70%) and testing (30%) subsets.
3. Fit TF-IDF on the training texts and generate TF-IDF features.
4. Generate SBERT embeddings for the training texts.
5. Concatenate TF-IDF and SBERT features into a unified hybrid representation.
6. Fit Robust Scaler on the training hybrid features and transform them.
7. Fit Truncated SVD on the normalized training features and project them into a lower-dimensional space.
8. Apply the fitted TF-IDF, Robust Scaler, and Truncated SVD transformations to the validation and test subsets without re-fitting.
9. Train the deep learning classifiers on the processed training data using class weights computed from the training labels to mitigate class imbalance.
10. Evaluate the trained models on the held-out test set using accuracy, precision, recall, F1-score, and ROC-AUC.

(1) 1D-CNN Model

The 1D-CNN technique applies to sequential data, in contrast to 2D and 3D-CNN algorithms, since it processes information in a single direction, yielding superior performance in text categorization through model architecture. Furthermore, 1D-CNN based log anomaly detection model that employs less variables than the models inside the present neural network family [20].

To identify discriminative patterns in the sequence of log representations, the (1D-CNN) architecture is used as shown in Figure 2. The model is a hybrid feature vector, which includes TF-IDF statistics value, as well as SBERT semantic embeddings, and hence utilizes both term significance and contextual significance in log sequences. Such representation-level fusion can be considered among the primary components of the suggested framework, as the study is not based on the classifier architecture only, but on complementary statistical and semantic information to use before classification. This first convolutional layer has 128 filters and a kernel size of 3 followed by ReLU activation. This layer adapts local patterns and local relationships between the input sequence. Following Batch Normalization and Spatial Dropout rate = 0.2 prevents feature co-adaptation and prevents overfitting, leading to improved training stability and convergence. Another 1D convolutional layer with 256 filters with the same kernel size enhances the ability of the model to pick intricate representations. This is also preceded by Batch Notice and Spatial Dropout which also offer regularization and resilience. A Global Max Pooling layer then downsizes the dimensionality of the feature maps, but does not lose the most salient characteristics. The resulting merged features are then run through two fully connected layers of 128 and 64 neurons respectively using ReLU activation. Every dense algebra is additionally regularized by dropout (dropout rate = 0.3) to prevent overfitting. The network is ended by one output value stemming out of a dense layer triggered by a sigmoid function thus approximating the likelihood of an abnormal log sequence in binary classification. Using the following equation:

$$\sigma = \frac{1}{1+ e^{-x}} \qquad (4)$$

Where x input value, e is Euler number nearly 2.718.

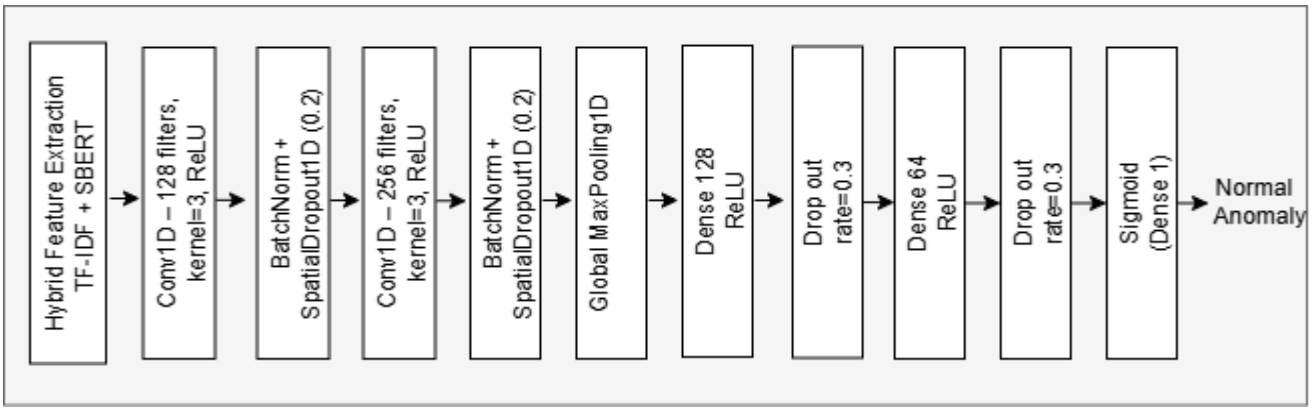


Figure 2: CNN Architecture.

(2) Bi-LSTM Model

Bi-LSTM is an extension of the conventional LSTM network. In contrast to traditional LSTM networks that process sequences unidirectional, Bidirectional LSTMs (Bi-LSTMs) facilitate information flow in both forward and backward directions, hence enhancing their capacity to capture contextual information. Bi-LSTMs are very successful for tasks that need comprehension of both past and future contexts

Figure 3 shows the Bi-LSTM architecture, which has been designed to capture temporal dependencies in two directions for the analysis of sequential and contextual system log data. Sentence-BERT (SBERT) embeddings: Sentence-Bert embeddings are used to transform the log messages into vectors of fixed length, which maintain the contextual semantics of the log message and is fed to a 64-unit Bi-LSTM layer.

Within this layer, two parallel sub-networks process the sequence in forward and backward temporal order simultaneously so that the model can detect anomalous behaviours with high reliability by exploiting dependencies across both past and future contexts. Batch Normalization is applied after the first Bi-LSTM layer to improve training stability and reduce internal covariate shift.

Subsequently, a second Bi-LSTM layer with 32 units is used to refine the temporal feature representation by learning higher-level sequential patterns from the normalized output. Another Batch Normalization layer is then introduced to further accelerate convergence and improve generalization.
The resultant temporal features are projected to a fully connected Dense layer with 128 neurons activated by the Rectified Linear Unit (ReLU) function to support nonlinear transformation and feature integration. A dropout layer with rate = 0.3 is added to reduce overfitting. This is followed by another Dense layer with 64 neurons, again regularized with dropout. Finally, a Sigmoid activation function is used in the output layer to generate a single probability value for binary classification of log sequences into normal or anomalous classes.

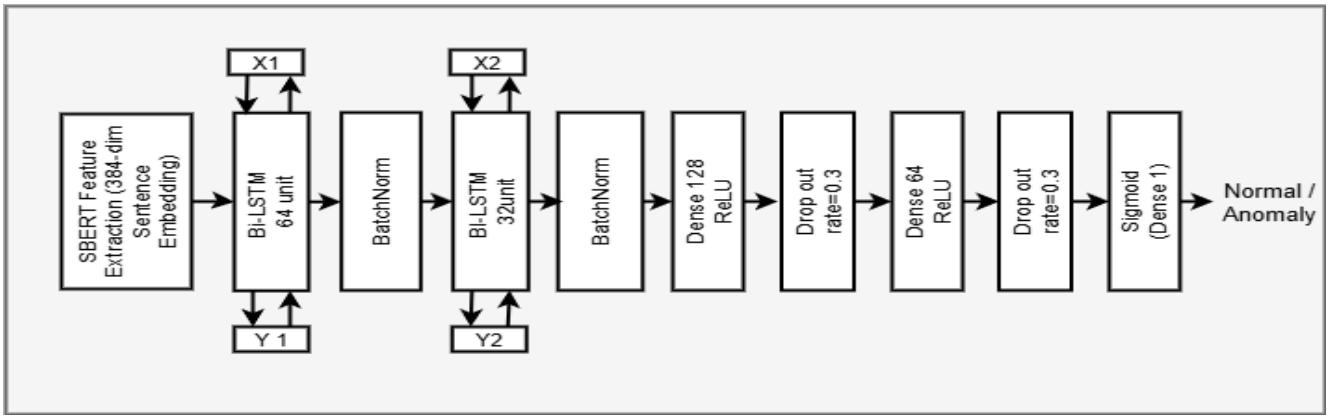


Figure 3: Bi-LSTM Architecture.

(3) Hybrid CNN-Bi LSTM Model

Figure 4 illustrates the design of the CNN-BiLSTM model intended for implementation in the proposed study. We trained the CNN-BiLSTM model as a hybrid architecture that integrates the two previously described individual models using the same parameter settings.

Each encoded log sequence is processed according to the dataset representation. The preprocessed features are passed from the embedding layer to the pooling layer, resulting in a fixed-length feature set similar to that of the CNN model. These feature sets are then fed into a Bi-LSTM model with 128 hidden units to compute contextual weights for the vectorized log sequence.

Finally, the features derived from the integrated models are projected through a fully connected layer, and the final classification into normal or anomalous classes is performed using the sigmoid output layer.

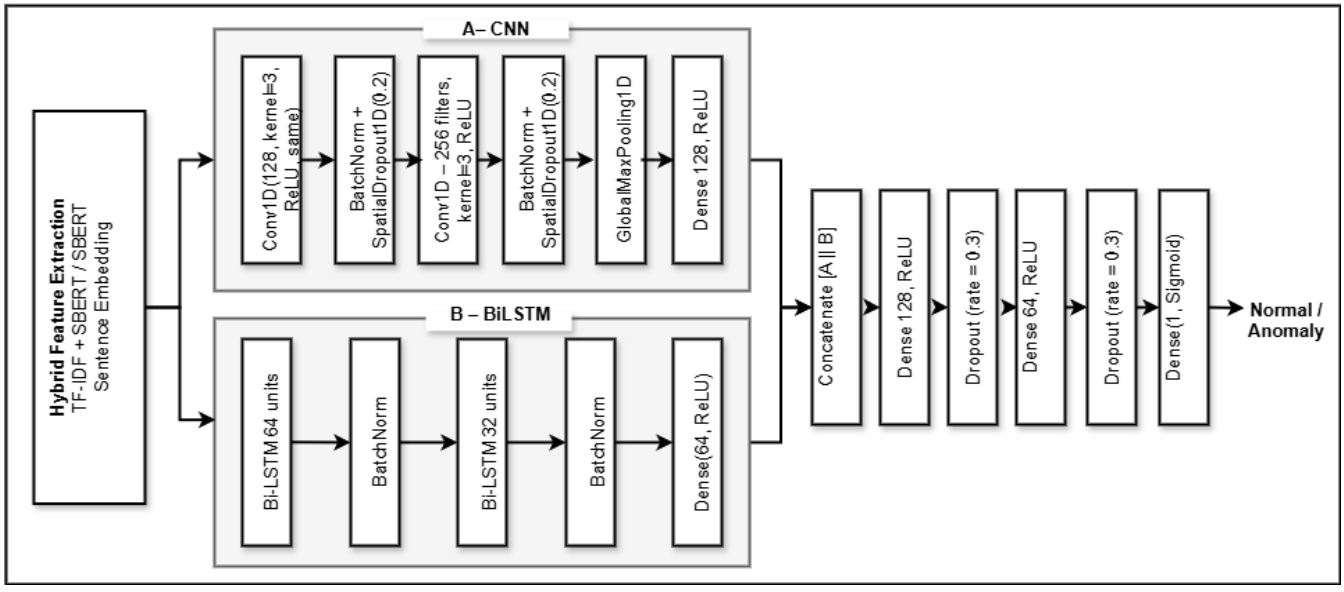


Figure 4: Hybrid CNN & Bi-LSTM Structure.

4. Results and Discussion

This approach was implemented using Python 3.13 and TensorFlow 2.x on a Windows-based system equipped with an Intel(R) Core (TM) i7-8850H CPU @ 2.60 GHz and 32 GB of RAM. The network was pre-trained on two datasets: HDFS [21] and BGL [15], containing 575,061 and 4747963 log messages, respectively. The distribution details are shown in Table 1.

Table 1: Details of the split HDFS and BGL datasets.

Class	HDFS dataset			BGL dataset		
	No. samples	70%Training	30%Testing	No. samples	70%Training	30%Testing
Normal	558223	390756	167467	4399503	3079652	1319850
Anomaly	16838	11787	5051	348460	243922	104538

We employ precision, recall, F1-score, and ROC curve to evaluate the effectiveness of the system for detecting anomalies in log messages. As shown in the following equations:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$f1 - score = \frac{2 \times (precision \times Recall)}{precision + Recall} \quad (8)$$

$$True\ Positive\ Rate = \frac{TP}{TP + FN} \quad (9)$$

$$True\ Negative\ Rate = \frac{TN}{TN + FP} \quad (10)$$

TP, TN, FP, and FN are the abbreviations that refer to the number of true positives, true negatives, false positives, and false negatives, respectively.

The testing confusion matrices of the CNN, Bi-LSTM, and hybrid CNN-BiLSTM models on the HDFS dataset are shown in Figure 5. Overall, the hybrid CNN-BiLSTM model exhibits the best classification behaviour among the evaluated models, with fewer misclassifications and a better balance between correctly detected normal and anomalous samples. These results indicate that the hybrid architecture is more effective in detecting anomalous events while reducing classification error.

Figure 5: The confusion matrix illustrating the experimental results of each model on the HDFS dataset.

Similarly, the testing confusion matrices on the BGL dataset, shown in Figure 6, further support the robustness of the proposed hybrid model. Compared with the individual CNN and Bi-LSTM models, the hybrid CNN-BiLSTM model shows better overall discrimination between normal and anomalous samples, indicating stronger generalization capability on large-scale log data.

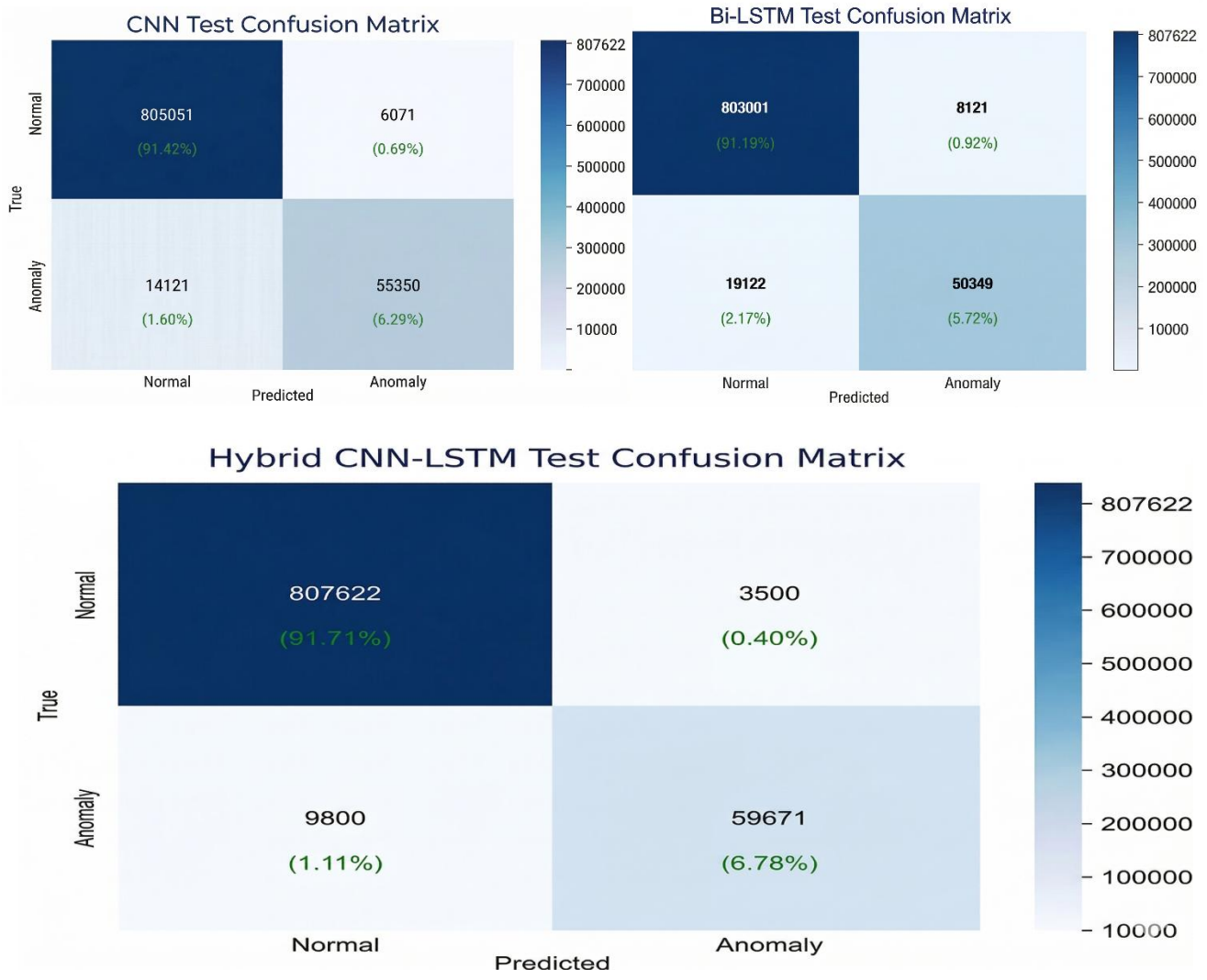


Figure 6: The confusion matrix illustrating the experimental results of each model on the BGL dataset.
(The confusion matrix is reported for the test subset after preprocessing, not for the raw dataset statistics)

In general, the consistent decrease in false positive and false negative in both datasets confirms the more reliable recognition of anomalous instances using the hybrid CNN-BiLSTM model. These findings show the generalization ability of the model and validate the usefulness when applied to different large-scale log datasets.

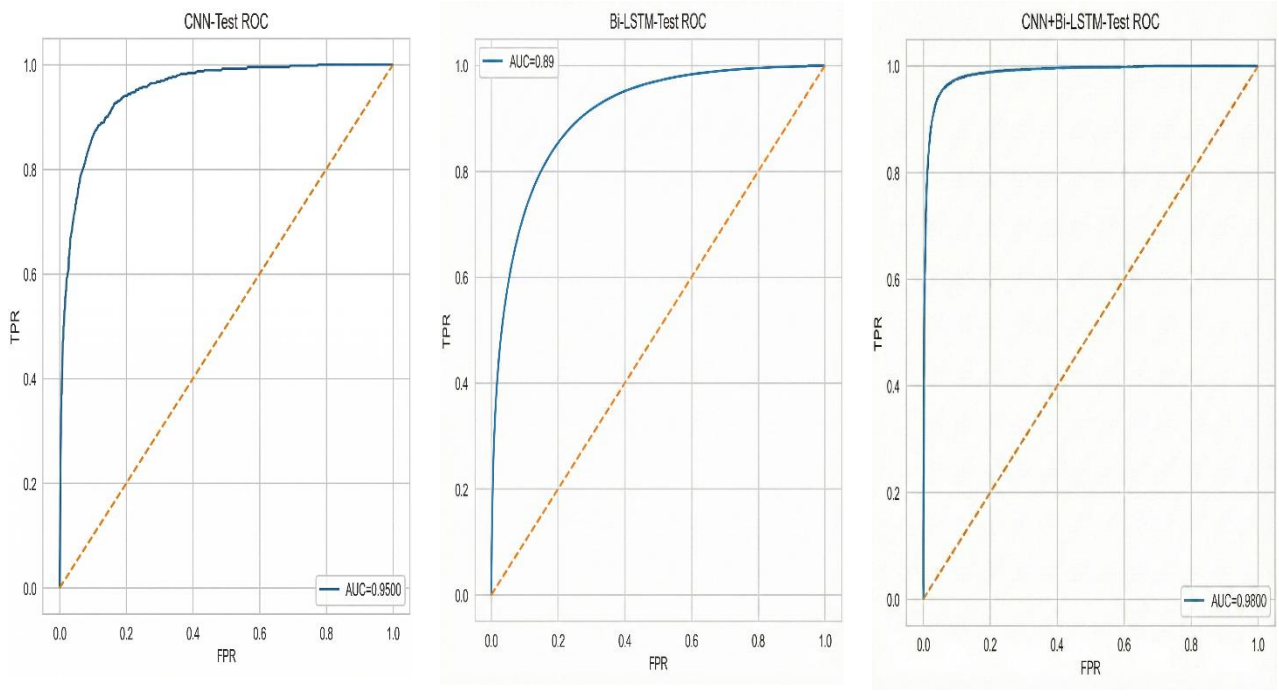
Table 2 is used to compare the performance of CNN, Bi-LSTM, and CNN-BiLSTM models based on the HDFS and BGL datasets using Accuracy, Precision, Recall, and F1-Score metrics. The CNN-BiLSTM hybrid model is the most effective model with both datasets, having the highest values for Accuracy and F1-Score, which reflects a good balance between accuracy and recall.

In the HDFS dataset, the hybrid model is definitely better than the individual models, especially in reducing false alarms and achieving a high detection rate. Similarly, on BGL data, it showed that it was more stable and performed better even with the difference in data characteristics. These results confirm the generalizability and high efficiency of the hybrid model in identifying anomalies in different environments.

Table 2: Evaluation of model efficacy across various datasets.

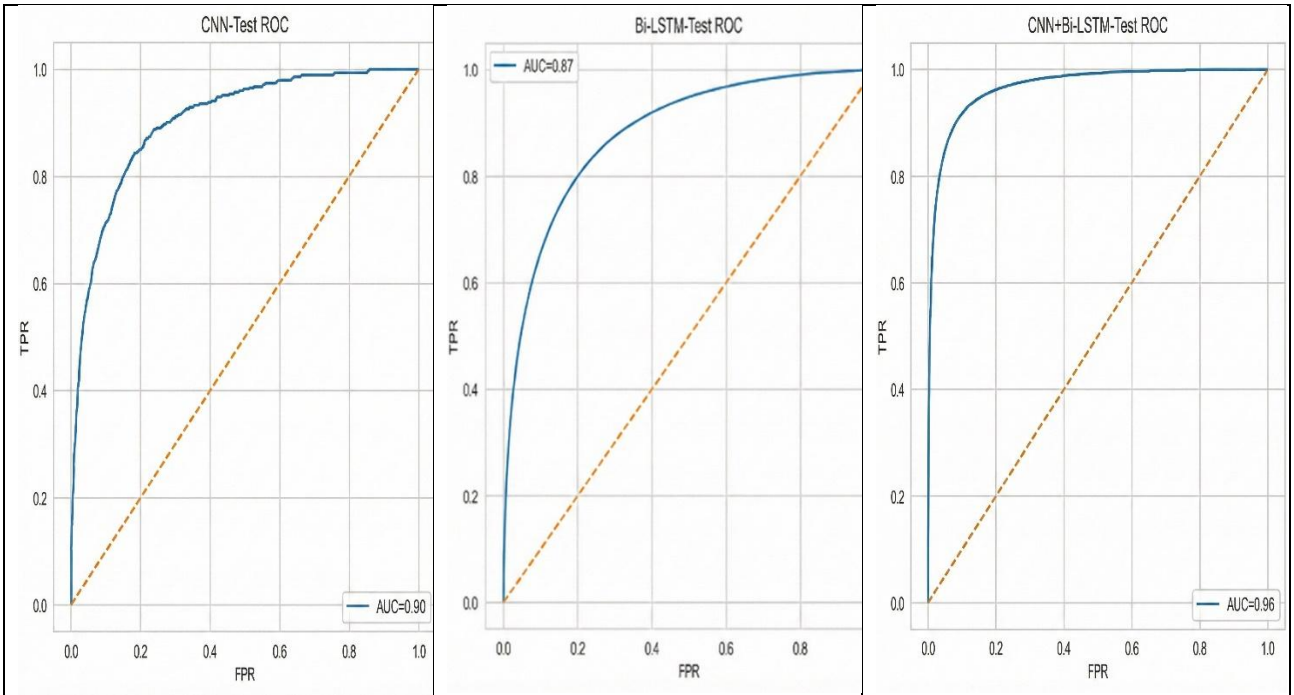
Model	HDFS dataset					BGL dataset				
	Accuracy	Precision	Recall	ROC-AUC	F1-Score	Accuracy	Precision	ROC-AUC	Recall	F1-Score
CNN	0.989	0.968	0.950	0.950	0.959	0.977	0.901	0.900	0.797	0.846
Bi-LSTM	0.987	0.942	0.885	0.890	0.913	0.969	0.861	0.870	0.725	0.787
Hybrid	0.997	0.969	0.955	0.980	0.962	0.983	0.936	0.960	0.847	0.889

A common metric for assessing the efficacy of classification models is the ROC AUC ("Receiver Operating Characteristic Area Under the Curve"). It delineates the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) across various categorization thresholds, serving as a critical metric of model efficacy. The ROC AUC value spans from 0 to 1, with values approaching 1 signifying superior model performance. Figure 7 and 8 illustrates the graph of the ROC curve. The x-axis of the graph represents the FPR. The y-axis of the graph represents TPR for each model across HDFS and BGL datasets.



(a) CNN Model **(b) Bi-LSTM Model** **(c) Hybrid CNN-BiLSTM Model**

Figure 7: The ROC Curve graph of each model on the HDFS dataset: (a) CNN Model, (b) Bi-LSTM Model, (c) Hybrid CNN-BiLSTM Model.



(a) CNN Model	(b) Bi-LSTM Model	(c) Hybrid CNN-BiLSTM Model

Figure 8: The ROC Curve graph of each model on the BGL dataset: (a) CNN Model, (b) Bi-LSTM Model, (c) Hybrid CNN-BiLSTM Model.

The figures show that the performance of the hybrid CNN & BiLSTM model is the best overall performance and achieves an AUC of 0.98 on the HDFS dataset and 0.96 on the BGL dataset. These results show a superior performance in separating the anomalous from the normal classes by a high true positive rate (TPR) and a low false positive rate (FPR).

The CNN model also has a competitive performance, achieving an AUC of 0.95 on the HDFS dataset and 0.90 on the BGL dataset. On the contrary, the performance of the Bi-LSTM model is relatively low and the AUC values are 0.89 and 0.87 on the HDFS and BGL datasets, respectively.

Overall, the superiority of the hybrid CNN-BiLSTM across both datasets suggests that it is robust.

The comparison between the proposed hybrid CNN-BiLSTM and related work, as shown in Table 3, demonstrates the performance of various state-of-the-art anomaly detection techniques applicable to log data. Earlier approaches, such as DeepLog [10] and the attention-based Bi-LSTM model [13], achieved competitive results, providing evidence of the effectiveness of recurrent neural architectures in capturing sequential log patterns. Methods based on self-attention mechanisms [11] and transformer-based architectures [14] further improved the modeling of long-range dependencies, while hybrid approaches such as SOM combined with BERT encoders and autoencoders [15] offered robust semantic feature extraction. Therefore, the novelty of the present study does not lie in the mere combination of CNN and Bi-LSTM, since similar hybrid architectures have been explored in prior studies. Rather, the main contribution lies in integrating statistical TF-IDF features with semantic SBERT embeddings within a unified framework, then employing a hybrid CNN-BiLSTM classifier to capture both local structural patterns and bidirectional contextual dependencies in log sequences. In addition, the proposed framework is experimentally validated on two benchmark datasets, HDFS and BGL, under the same evaluation setting. In comparison, the proposed Hybrid CNN-BiLSTM model shows strong performance with 0.9978 accuracy, 0.9698 precision, 0.9553 recall, and an F1 score of 0.9625 on the HDFS dataset. Similarly, it achieves 0.9833 accuracy, 0.936 precision, 0.847 recall, and an F1 score of 0.8894 on the BGL dataset.

Table 3: Comparison of the proposed Hybrid CNN-BiLSTM Model with relevant literature. A '_' occurs, indicating that comparable experimental results are unavailable.

Model	Dataset	Methodology	Accuracy	Precision	Recall	F1
[10]	HDFS	LSTM-based sequence model (DeepLog)	_	0.996	0.978	0.988
[11]	Special dataset	Self-attention neural network	0.890	0.90–0.99	0.90–0.99	_
[12]	HDFS	LSTM and Template2vec	_	_	_	0.950
	BGL		-	-	-	0.960
[13]	HDFS	attention-based Bi-LSTM	-	0.962	0.968	0.952
[14]	HDFS	Transformer Encoder + Evidential Deep Learning (EDL)	-	0.900	0.923	0.914
	BGL		-	-	-	0.985
[15]	HDFS	SOM + BERT Encoder + Autoencoders (Hybrid)	0.950	0.940	0.96	0.950
	BGL		-	0.970	0.94	0.960
[17]	HDFS	CNN-LSTM	_	_	_	0.988
Our	HDFS	Hybrid CNN-BiLSTM	0.997	0.969	0.9553	0.962
	BGL		0.983	0.936	0.847	0.889

This can be explained by the system's dual-stage structure, which led to the observed improvement. The CNN layers can identify local structural dependencies in log sequences, while the Bi-LSTM component can preserve bidirectional temporal dependencies. The complementary interaction between the two stages of the model allows it to detect anomalous patterns with high sensitivity. Moreover, the fusion process combining statistical (TF-IDF) and semantic (SBERT) feature representations yields more discriminative feature vectors, thereby increasing the model's ability to distinguish between normal and anomalous events. Therefore, the contribution of the proposed method lies in this unified feature-fusion and hybrid-learning, where, the suggested model provides an effective framework for log anomaly detection on the HDFS and BGL datasets and shows that semantically enriched representations combined with hybrid deep-learning stages can outperform standalone or single-stage models under the evaluated setting.

Computational Efficiency and Practical Concerns: In the real-world implementation, computational efficiency is a matter of concern to log anomaly detection systems. The detection effectiveness that was mainly studied in the present study was related to accuracy, precision, recall, F1-score, and ROC-AUC. Even though the hybrid approach has a superior predictive performance, it is likely to be more expensive to compute than single-model baselines because it combines both the feature representation and the sequential deep learning parts. In the current study, a detailed comparison of training time, inference latency, and the number of parameters was not provided, and is an essential direction of further research that would allow practicing in real time.

Limitations: Despite the encouraging results, the present study has several limitations. First, the evaluation was conducted on two benchmark datasets only, namely HDFS and BGL, and therefore the generalizability of the proposed framework to other log sources and operational environments still requires further investigation. Second, the current study focused on aggregate performance metrics and did not include a detailed qualitative analysis of misclassified log instances or failure cases. In practice, such failure cases may arise when log messages are ambiguous, highly noisy, or semantically similar across normal and anomalous events. These aspects should be examined in future work to further improve robustness, interpretability, and deployment readiness.

5- Conclusion

This study introduced a dual hybrid framework that combines statistical and semantic feature representation with a deep learning–based classification model for log anomaly detection. By integrating TF-IDF with SBERT embeddings, the proposed method captures both term significance and contextual meaning within unstructured log data. The hybrid CNN-BiLSTM architecture achieved the highest performance across all evaluation metrics, including accuracy, F1-score, and ROC–AUC, demonstrating superior capability in distinguishing normal and anomalous log sequences. These findings confirm that integrating convolutional and bidirectional recurrent learning enhances detection reliability. In conclusion, our research enhances reproducibility in cybersecurity, particularly for log data and intrusion detection, thereby augmenting the capacity to base future studies on existing findings.

References

[1] Z. T. M. Al-Ta'i and S. M. Sadoon, "Visual cryptography based on chaotic logistic map in multi-cloud," in AIP Conference Proceedings, 2024, vol. 3097, no. 1.

[2] M. Alabadi and Y. Celik, "Anomaly Detection for Cyber-Security Based on Convolution Neural Network: A survey," HORA 2020 - 2nd Int. Congr. Human-Computer Interact. Optim. Robot. Appl. Proc., no. January, 2020, doi: 10.1109/HORA49412.2020.9152899.

[3] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," Mach. Learn. with Appl., vol. 12, no. March, p. 100470, 2023, doi: 10.1016/j.mlwa.2023.100470.

[4] R. Foorthuis, "On the nature and types of anomalies: A review of deviations in data," SN Computer Science, vol. 12, no. 4, 2021, doi: 10.1007/s41060-021-00265-1.

[5] C. Sánchez-Zas, X. Larriva-Novo, V. A. Villagrá, M. S. Rodrigo, and J. I. Moreno, "Design and Evaluation of Unsupervised Machine Learning Models for Anomaly Detection in Streaming Cybersecurity Logs," Mathematics, vol. 10, no. 21, 2022, doi: 10.3390/math10214043.

[6] Z. A. Khan, D. Shin, D. Bianculli, and L. C. Briand, "Impact of log parsing on deep learning-based anomaly detection," Empirical Software Engineering, vol. 29, no. 6, 2024, doi: 10.1007/s10664-024-10533-w.

[7] M. Goldstein and S. Uchida, "Behavior Analysis Using Unsupervised Anomaly Detection," 10th Jt. Work. Mach. Percept. Robot., no. October, 2014.

[8] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," IEEE Access, vol. 9, pp. 78658–78700, 2021, doi: 10.1109/ACCESS.2021.3083060.

[9] T. Rajendran, N. Mohamed Imtiaz, K. Jagadeesh, and B. Sampathkumar, "Cybersecurity Threat Detection Using Deep Learning and Anomaly Detection Techniques," 2024 Int. Conf. Knowl. Eng. Commun. Syst. ICKECS 2024, vol. 1, pp. 1–7, 2024, doi: 10.1109/ICKECS61492.2024.10617347.

[10] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, 2017, pp. 1285–1298.

[11] Z. Liu, T. Qin, X. Guan, H. Jiang, and C. Wang, "An integrated method for anomaly detection from massive system logs," IEEE Access, vol. 6, pp. 30602–30611, 2018.

[12] W. Meng et al., "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs.," in IJCAI, 2019, vol. 19, no. 7, pp. 4739–4745.

[13] C. Zhang et al., "LayerLog: Log sequence anomaly detection based on hierarchical semantics," Appl. Soft Comput., vol. 132, p. 109860, 2023, doi: 10.1016/j.asoc.2022.109860.

[14] Y. Duan et al., "LogEDL: Log Anomaly Detection via Evidential Deep Learning," Appl. Sci., vol. 14, no. 16, pp. 1–18, 2024, doi: 10.3390/app14167055.

[15] A. Aziz and K. Munir, "Anomaly Detection in Logs Using Deep Learning," IEEE Access, vol. 12, no. November, pp. 176124–176135, 2024, doi: 10.1109/ACCESS.2024.3506332.

[16] R. Jassim, "Review of Computer Engineering Research Artificial intelligence methods for identification of ADHD in children based on EEG signals Keyword s," vol. 12, no. 2, pp. 80–93, 2025, doi: 10.18488/76. v12i2.4217.

[17] A. Falini, "A review on the selection criteria for the truncated SVD in Data Science applications," J. Comput. Math. Data Sci., vol. 5, p. 100064, 2022.

[18] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," Expert Systems with Applications, vol. 73, pp. 220–239, 2017.

[19] J. da Silva Freitas Junior and P. H. Pisani, "Performance and model complexity on imbalanced datasets: An experimental comparison of cost-sensitive and resampling methods," Proceedings of Machine Learning Research, vol. 183, 2022.

[20] "Utility analysis about log data anomaly detection based on federated learning," Applied Sciences, 2023.

[21] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," in Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles, 2009, pp. 117–132.