International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Anomaly Detection in Distributed File System Logs Using Hybrid Machine Learning and Deep Learning Models

Dhruv Mishra, Chirag Ananda Kumar, Chatura J S, Aliasgar Abbas Ringnodwala and Hema M S
Department of Computer Science and Engineering
RV Institute of Technology and Management
Bengaluru – 560076, Karnataka, India

Abstract—This paper proposes a hybrid anomaly detection framework for Distributed File System (DFS) log analysis, combining statistical machine learning with deep learning sequence models. Rather than relying on a single classifier, the system integrates Random Forest, Linear Support Vector Machine (SVM), Bidirectional Long Short-Term Memory (BiLSTM), and DeepLog within a weighted soft-voting ensemble. Class imbalance is addressed through the Synthetic Minority Over-sampling Technique (SMOTE), applied to classical learners prior to training. Precision-Recall Area Under the Curve (PR-AUC) is adopted as the primary evaluation metric, given its suitability for imbalanced classification tasks. The dual-modality design captures both event-frequency patterns via Bag-of-Words representations and temporal ordering dependencies via sequential models. Experiments on the public HDFS LogHub dataset show the ensemble achieves a PR-AUC of 0.767 and perfect precision, while Random Forest attains the highest F1-score and a ROC-AUC of 0.953. The framework provides a reliable and interpretable approach to log-based anomaly detection in large-scale distributed systems.

Keywords—*System log file Analysis; Anomaly Detection; Hybrid Machine Learning; Deep Learning; Log-based Monitoring; Random Forest; BiLSTM(Ensemble); DeepLog(Ensemble); SMOTE (Class Imbalance); Precision-Recall AUC (PR-AUC).*

I. Introduction

Large-scale distributed computing environments, such as Hadoop Distributed File System (HDFS), continuously generate high-volume log data that records system behavior, resource utilization, and fault conditions. Manual inspection of such logs is operationally infeasible at scale, and rule-based approaches remain inherently limited in their adaptability to evolving system dynamics. Machine learning (ML) and deep learning (DL) methods have therefore emerged as practical alternatives for automated log anomaly detection, offering the capacity to infer behavioral patterns directly from raw log data [1]–[3].

Despite this progress, log anomaly detection remains challenging for three principal reasons. First, anomalous events constitute a small minority of log entries, creating severe class imbalance. Second, log data is unstructured and high-dimensional, requiring effective feature representations prior to model training. Third, anomalies in distributed systems frequently manifest through temporal dependencies among sequential events rather than through isolated log entries alone. Addressing these challenges requires a framework that unifies complementary modeling paradigms. This study presents a hybrid ensemble that combines Random Forest and Linear Support Vector Machines (SVM) with BiLSTM and DeepLog through a weighted soft-voting mechanism, augmented by SMOTE for class-imbalance mitigation. Sequential deep learning components are incorporated as ensemble members rather than standalone detectors, enabling the framework to capture both statistical and contextual anomaly signals without sacrificing individual model interpretability.

II. Methodology

A. Dataset and Preprocessing

Experiments are conducted on the HDFS log dataset from the open-access LogHub repository [7], which contains structured log records collected from a Hadoop distributed storage cluster. Each log entry is tagged with a Block ID that identifies the corresponding HDFS block operation. Log entries are grouped by Block ID to form block sessions, where each session constitutes the complete sequence of log events generated during the lifecycle of a single block operation. This session-level grouping serves as the fundamental unit of analysis.

Each block session is represented as an ordered sequence of Event IDs with a binary ground-truth label—normal or anomalous—derived from expert annotations provided in the dataset. Two complementary data representations are constructed to support the different modeling approaches. For frequency-based classifiers, log messages within each session are concatenated into a single text string and transformed using a Bag-of-Words (BoW)

representation. For sequence-aware models, the ordered Event ID sequence is preserved and used directly as input.

B. Exploratory Data Analysis

Preliminary analysis reveals a pronounced class imbalance: anomalous sessions constitute approximately 4% of the total dataset, with 7,627 normal sessions and 313 anomalous sessions. This distribution confirms that standard accuracy metrics are insufficient for evaluating model performance in this setting, motivating the adoption of PR-AUC as the primary evaluation criterion.

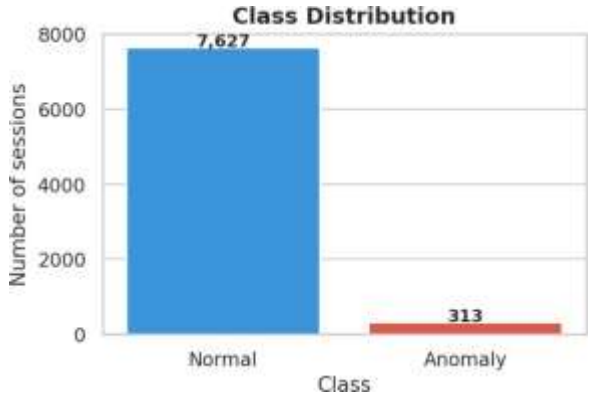


Fig. 1. Class distribution of Distributed Files System log sessions (Normal: 7,627; Anomaly: 313).

1) Session Length Analysis:

Session length analysis shows that normal sessions follow a concentrated distribution with low variance, while anomalous sessions exhibit substantially greater variability. This divergence in event count per session suggests that anomalous block operations are associated with atypical execution patterns, either terminating prematurely or extending well beyond the normal range.

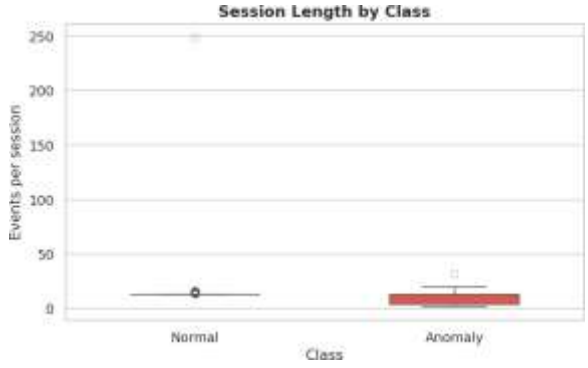


Fig. 2. Session length by class: normal sessions are concentrated, while anomalous sessions exhibit higher variance.

2) Event Frequency Patterns:

Examination of per-event occurrence rates reveals that certain Event IDs appear with disproportionately higher frequency in anomalous sessions than in normal ones. This systematic frequency skew supports the use of count-based feature representations as discriminative inputs for classical classifiers and indicates that anomalous block operations are consistently associated with specific event types rarely observed during normal execution.

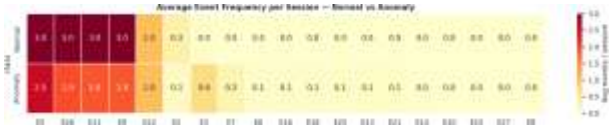


Fig. 3. Average event frequency per session — Normal vs. Anomaly.

3) Distribution of Events per Session:

Histogram analysis of events per session confirms that anomalous sessions deviate markedly from the modal behavior of normal sessions. While most normal sessions cluster within a narrow range of event counts, anomalous sessions exhibit a long-tailed distribution extending well beyond the normal range. This finding reinforces the utility of frequency-based features and motivates combining them with sequential modeling.

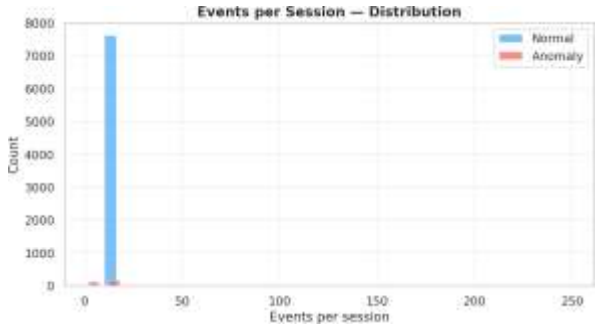


Fig. 4. Distribution of events per session, separated by class.

C. Feature Engineering

Raw log messages are transformed into numerical feature vectors using a Bag-of-Words representation. Each session is treated as a text document, and unigram and bigram token frequencies are computed across all messages within the session. To control dimensionality, the vocabulary is restricted to the 1,500 most frequent terms, reducing the risk of overfitting to rare tokens while retaining the most discriminative lexical patterns.

To address the class imbalance identified during exploratory analysis, SMOTE is applied to the training split prior to model fitting. SMOTE generates synthetic minority-class instances by interpolating between existing anomalous samples in the feature space, improving classifier exposure to rare event patterns. This over-sampling strategy is applied exclusively to classical ML models; sequential components handle imbalance through separate mechanisms described in Section II-E.

D. Models Used

Three detection configurations are evaluated: a standalone Random Forest classifier, a standalone Linear SVM classifier, and the proposed ensemble model combining all four components.

1) Random Forest (Primary Model):

The Random Forest classifier is configured with the following settings:

- 300 estimators
- Class-balanced training
- Captures frequency-based anomalies effectively

A Random Forest comprising 300 decision trees is trained on BoW feature vectors with class-balanced sample weighting. The ensemble of independently trained trees provides robustness to noisy features and reduces overfitting. The architecture is well-suited to high-dimensional sparse BoW inputs, as each tree is trained on a random feature subset. Feature importances derived from Gini impurity scores are computed post-training to identify the most discriminative log tokens.

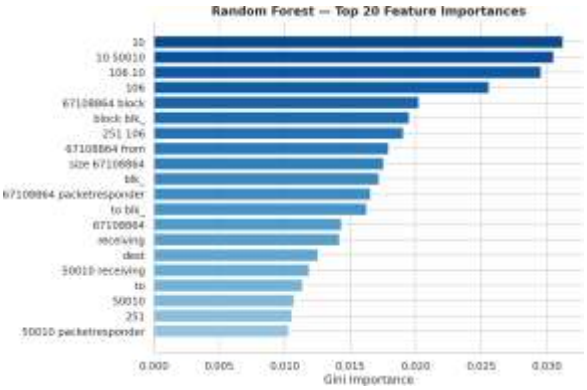


Fig. 5. Random Forest — Top 20 feature importances by Gini score.

2) Linear SVM:

The Linear SVM classifier employs the following approach:

- Margin-based classifier
- Calibrated probabilities
- Strong recall performance

A Linear SVM classifier with calibrated posterior probabilities is trained on the same BoW feature vectors. Probability calibration is performed via Platt scaling (sigmoid fitting), mapping SVM decision scores to well-calibrated class probability estimates. This step enables the Linear SVM to participate in soft-voting alongside probabilistic classifiers. The model is particularly effective in high-dimensional sparse feature spaces and demonstrates strong recall on the anomaly class.

3) Ensemble Model (Proposed):

The proposed ensemble combines four component models through weighted soft-voting: Random Forest (weight 0.4), BiLSTM (weight 0.4), and DeepLog (weight 0.2). Weights were determined empirically on a held-out validation set. By integrating frequency-based and sequence-aware representations, the ensemble is designed to capture complementary anomaly signatures that individual models may not detect in isolation.

E. Sequential Components (Ensemble Only)

1) BiLSTM(Ensemble):

The BiLSTM component processes the ordered Event ID sequence within each session to capture long-range dependencies across the full event history. The bidirectional architecture processes each sequence in both forward and reverse temporal directions, integrating contextual information from both preceding and subsequent events at each time step. To address class imbalance during training, focal loss is employed instead of standard binary cross-entropy, down-weighting easily classified normal events and focusing training on hard anomalous samples.

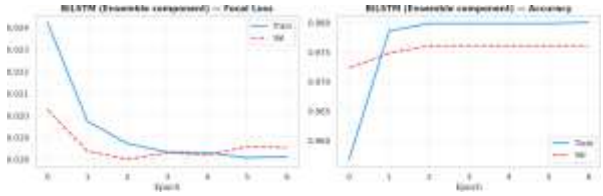


Fig. 6. BiLSTM (Ensemble component) — Focal Loss and Accuracy training curves.

2) DeepLog(Ensemble):

DeepLog [4] models log sequential structure using a sliding-window LSTM that predicts the next Event ID given a fixed-length history of preceding events. At inference time, a session is flagged as anomalous if the observed next event falls outside the set of top-k predicted candidates at any position in the sequence. This criterion is grounded in the observation that anomalous block operations involve event transitions deviating from patterns learned during normal operation, contributing a detection signal complementary to aggregate frequency statistics.

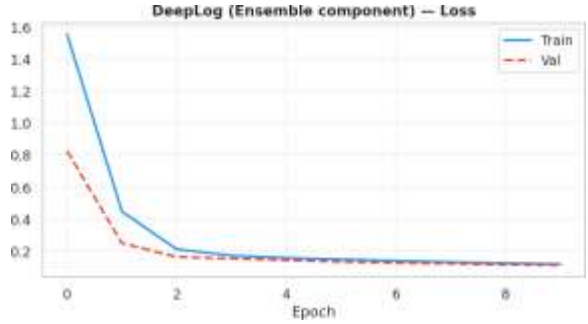


Fig. 7. DeepLog (Ensemble component) — Loss convergence curve over training epochs.

F. Evaluation Metrics

Given the substantial class imbalance, PR-AUC is adopted as the primary performance metric. It measures the area under the precision-recall curve across all classification thresholds, providing a more informative assessment than overall accuracy or ROC-AUC for imbalanced datasets. The no-skill baseline for PR-AUC is approximately 0.04, corresponding to the anomalous session ratio. Additional reported metrics include precision, recall, F1-score, and ROC-AUC, which together characterize each model's detection behavior across operating points.

III. Results

This section evaluates the anomaly detection performance of the proposed ensemble and baseline models on the HDFS test set. With approximately 4% of sessions being anomalous, standard accuracy metrics are unreliable; evaluation therefore focuses on PR-AUC, precision, recall, and F1-score, alongside analysis of confusion matrices, ROC curves, and threshold sensitivity.

A. Confusion Matrices

Random Forest achieves 1,524 correct calls on normal sessions with only 1 false positive, while 33 true anomalies are correctly detected and 30 are missed. Linear SVM flags 17 false positives but catches 39 true anomalies, showing a bias toward detection coverage. The Ensemble achieves perfect precision with zero false positives, catching 30 true anomalies but missing 33, reflecting a conservative decision threshold.

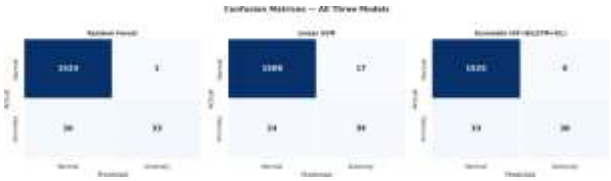


Fig. 8. Confusion matrices for all three models on the test set.

B. Overall Model Performance

All three models achieve overall accuracy in the range of 97–98%; however, this metric is an unreliable indicator given the class distribution. The Ensemble achieves the highest precision at 1.000, with Random Forest close at 0.970. Linear SVM leads in recall at 0.619, while the Ensemble achieves 0.476. Random Forest attains the highest F1-score of 0.680. For the primary PR-AUC metric, the Ensemble and Random Forest achieve approximately 0.767 each, while Linear SVM scores 0.707. All three configurations substantially outperform the no-skill baseline of 0.040.

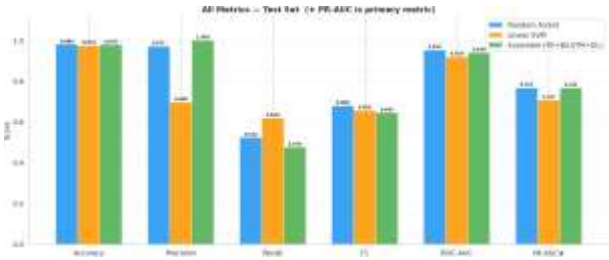


Fig. 9. All metrics comparison on the test set (PR-AUC is the primary metric).

C. ROC Curves

ROC curve analysis confirms the discriminative capability of each model across threshold settings. Random Forest achieves the highest ROC-AUC of 0.953, followed by the Ensemble at approximately 0.940 and Linear SVM at 0.920. These results indicate that all three models effectively separate anomalous from normal sessions across a range of operating conditions.

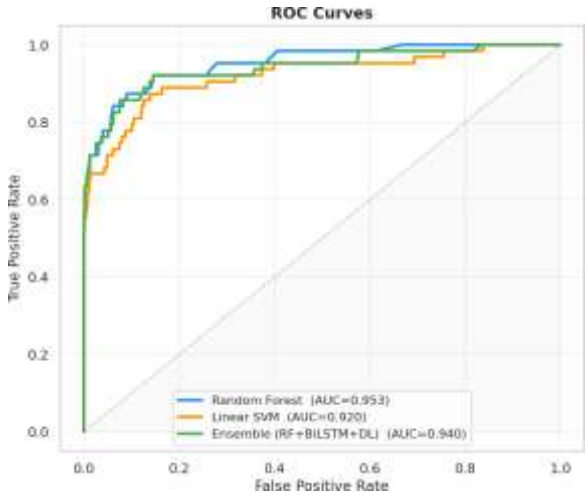


Fig. 10. ROC curves for all three models.

D. Precision-Recall Curves

Precision-recall curves provide the most diagnostic performance summary under class imbalance. Random Forest and the Ensemble achieve nearly identical PR-AUC scores of approximately 0.767, indicating that integrating sequential components does not reduce the precision-recall performance established by the Random Forest baseline. Linear SVM achieves a PR-AUC of 0.707. The pronounced gap between all models and the no-skill baseline (0.040) confirms the effectiveness of the feature engineering and modeling design.

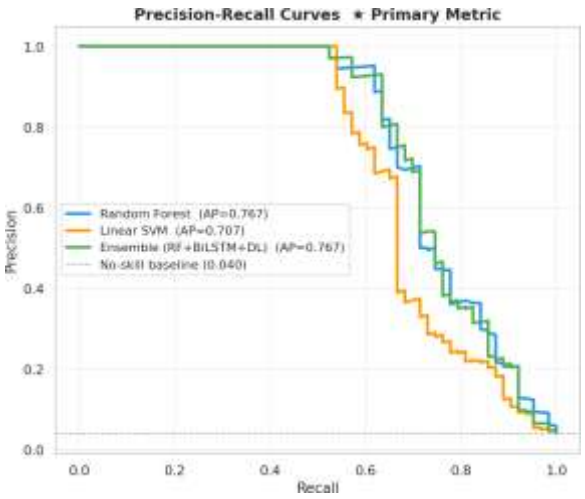


Fig. 11. Precision-Recall curves (primary metric). No-skill baseline at 0.040.

E. Recall vs. Threshold

Recall-versus-threshold analysis reveals qualitatively distinct sensitivity profiles across the three models. Random Forest maintains stable recall across a broad range of threshold values. The Ensemble exhibits a sharp decline in recall beyond a threshold of approximately 0.6, reflecting a concentration of predictive mass at high-confidence anomaly scores. Linear SVM shows a more gradual recall decay, sustaining moderate detection rates across a wider range of thresholds, which may be advantageous in deployments where false negatives carry high operational cost.

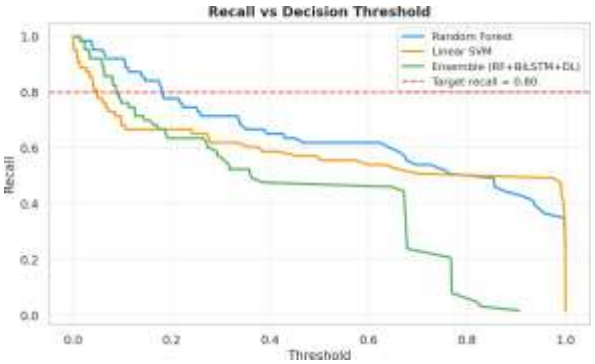


Fig. 12. Recall vs. decision threshold for all three models.

F. Precision vs. Recall Trade-off

At the tuned decision thresholds, the precision-recall trade-off profiles diverge substantially. The Ensemble achieves perfect precision at a recall of 0.476, making it appropriate for scenarios where false alarms carry high operational cost. Random Forest achieves near-perfect precision while recovering a marginally larger proportion of true anomalies. Linear SVM maximizes recall at the cost of reduced precision, making it most suitable for high-sensitivity monitoring where missed anomalies are more costly than false positives.

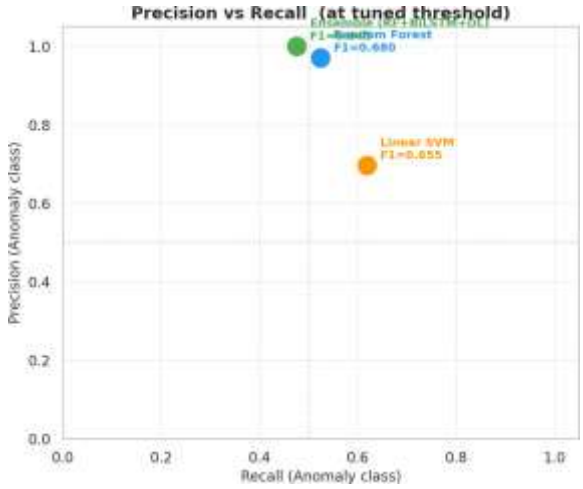


Fig. 13. Precision vs. Recall at the tuned decision threshold for each model.

G. Radar Chart Comparison

Radar chart visualization confirms that Random Forest achieves the most balanced performance profile across all evaluation dimensions, with no single metric substantially underperforming. The Ensemble demonstrates a clear specialization toward high-precision detection, while Linear SVM exhibits a comparative advantage in recall. Together, these profiles illustrate the complementary nature of the three detection configurations.

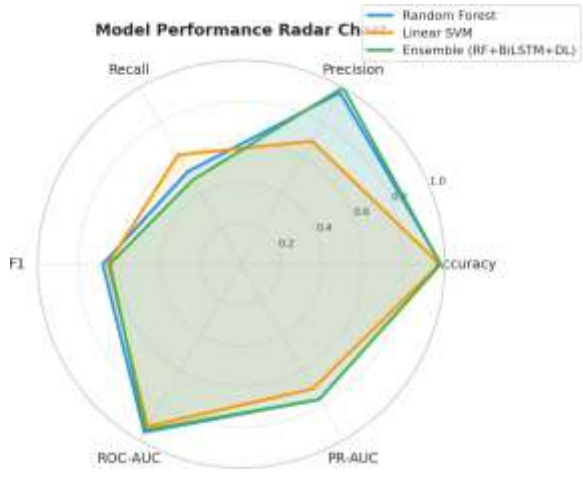
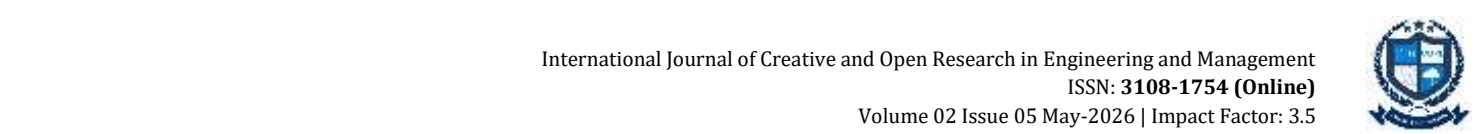


Fig. 14. Model performance radar chart across all evaluation metrics.

H. Metrics Heatmap

The metrics heatmap consolidates all performance data into a single comparative view. Random Forest achieves a PR-AUC of 0.7670, marginally exceeding the Ensemble at 0.7658. The Ensemble leads in precision at 1.0000, while



Linear SVM achieves the highest recall at 0.6190, confirming its advantage in detecting the greatest proportion of anomalous sessions.

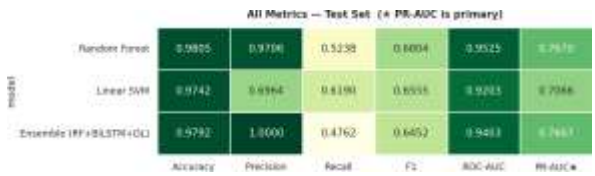


Fig. 15. Metrics heatmap — all models on the test set (PR-AUC is the primary metric).

I. Key Findings

Based on the experimental results, the following key observations are drawn:

- Random Forest achieves the best overall PR-AUC (~0.767) with a strong balance between precision and recall.
- Linear SVM achieves the highest recall and is most suitable when missing anomalies is critical.
- The Ensemble Model matches Random Forest in PR-AUC while capturing both frequency-based and sequential anomaly patterns, providing comparable performance while incorporating dual modeling insights.

IV. Conclusion

This paper presents a hybrid anomaly detection framework for Distributed File System logs that integrates classical machine learning and deep learning within a weighted ensemble architecture. The framework combines Random Forest and Linear SVM classifiers operating on Bag-of-Words frequency representations with BiLSTM and DeepLog sequence models that capture temporal event dependencies. SMOTE is applied to mitigate class imbalance for frequency-based classifiers, while focal loss addresses imbalance within the BiLSTM component. Sequential models are incorporated as ensemble members rather than standalone detectors, enabling the framework to benefit from both statistical and contextual anomaly signals.

The experimental results show that traditional models such as Random Forest and Linear SVM provide strong baseline performance, with Random Forest achieving the best balance between precision and recall. The ensemble model achieves comparable PR-AUC while incorporating both frequency-based and sequential insights, offering a more comprehensive anomaly detection framework. These findings highlight that combining different modeling paradigms can enhance coverage of diverse anomaly patterns, even when overall performance remains similar to strong individual baselines.

The proposed framework demonstrates reliable and interpretable anomaly detection by capturing both statistical and sequential characteristics of log data. Future work will investigate transformer-based architectures fine-tuned on log sequences, development of online anomaly detection mechanisms for streaming log ingestion, and systematic evaluation of framework generalizability across diverse distributed system environments including cloud-native and microservice architectures.

References

[1]R. Vaarandi, “A Data Clustering Algorithm for Mining Patterns from Event Logs,” in Proc. IEEE Workshop on IP Operations and Management (IPOM), 2003.

[2]M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, “A Novel Anomaly Detection Scheme Based on Principal Component Classifier,” in Proc. IEEE Foundations and New Directions of Data Mining Workshop, 2003.

[3]W. Xu, L. Huang, A. Fox, D. A. Patterson, and M. I. Jordan, “Detecting Large-Scale System Problems by Mining Console Logs,” in Proc. ACM SIGOPS Symposium on Operating Systems Principles (SOSP), 2009.

[4]M. Du, F. Li, G. Zheng, and V. Srikumar, “DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning,” in Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017.

[5]P. He, J. Zhu, Z. Zheng, and M. R. Lyu, “Drain: An Online Log Parsing Approach with Fixed Depth Tree,” in Proc. IEEE International Conference on Web Services (ICWS), 2017.

[6]K. Zhang, J. Xu, M. Min, et al., “LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs,” in Proc. International Joint Conference on Artificial Intelligence (IJCAI), 2019.

[7]J. Zhu, S. He, P. He, J. Liu, and M. R. Lyu, “LogHub: A Large Collection of System Log Datasets for AI-Driven Log Analytics,” arXiv preprint arXiv:2008.06448, 2020.

[8]Y. Alaca, E. Başaran, and Y. Çelik, “Enhancing Anomaly Detection in Large-Scale Log Data Using Machine Learning: A Comparative Study of SVM and KNN Algorithms with HDFS Dataset,” Empirical Software Engineering, 2024.