A Semi-Supervised Ensemble Framework for Anomaly Detection in Cybersecurity Logs

Harbi Mahmood Abbas[1], Ziyad Tariq Mustafa Al-Ta'i[1] and Hamza Aagela[2]
[1]*Department of Computer Science, College of Science, University of Diyala, 32001 Baqubah, Iraq*
[2]*Department of Electrical and Electronic Engineering, University of Huddersfield, HD1 3DH Huddersfield, United Kingdom*
scicompms242510@uodiyala.edu.iq, Ziyad1964tariq@uodiyala.edu.i, h.ym.aagela@hud.ac.uk

Keyword: Semi-Supervised, Ensemble Learning, Anomaly Detection, Cybersecurity Logs, Deep Learning.

Abstract: Anomaly detection is crucial in cybersecurity logs; nevertheless, system logs' extensive size and complexity render manual analysis impractical. Traditional supervised methods necessitate extensive labelled datasets, whereas unsupervised methods lack robustness. To address these difficulties, we proposed a novel semi-supervised framework for anomaly detection in cybersecurity logs. It employs a hybrid feature representation, deep learning, traditional models, and ensemble techniques. The framework has many critical layers: hybrid feature representation TF-IDF (sparse feature), SBERT (semantic feature), and statistical features. Anomaly detection employs an Auto-Encoder, a Bi-LSTM module, and two traditional models: an isolation forest and a one-class support vector machine. The outputs of these models are integrated using a two-layer approach: weighted averaging (soft voting) and stacking via a random forest optimizer. Experimental findings on the HDFS dataset demonstrate that this hybrid semi-supervised approach enhances detection accuracy, scalability, and robustness, offering an efficient method for enhancing cybersecurity via log-based anomaly detection.

1 INTRODUCTION

As organizations grow more dependent on digital infrastructure, the intricacy and magnitude of cybersecurity threats persistently escalate. Traditional security protocols frequently fail to identify advanced assaults, particularly new or concealed within legitimate traffic [1]. Emerging risks, such as denial-of-service (DoS) attacks and advanced persistent attacks (APTs), necessitate adaptive security systems that continuously learn and evolve.

Anomaly detection is essential to modern cybersecurity methods as it identifies abnormalities in network traffic or user activity behavior[2]. It is crucial for spotting emerging attack patterns, as it can detect both known and undiscovered threats, particularly when there is minimal or no prior information about a specific attack. Early detection of unusual behavior can avert significant damage and reduce potential risks [3].

Analyzing event and system-based logs through synthesizing many systems, software, and hardware is essential in wide internet networks. Log recordings, gathered from devices and software that ensure system security, frequently include evidence of assaults executed by malicious actors during or subsequent to an incident. Consequently, it is imperative to examine log records and discover anomalies arising from these traces to detect cyber-attacks [4].

The existing techniques for analyzing event and system-based logs can be categorized into three primary types: log key sequence based methods (such as, Deeplog [5] and logkey2vec [6]), log event count based methods (such as, SVM [7], PCA [8], LogClustering [9]) and log template based methods (like LogRobust and LogAnomaly [10]).

Those methods depend on log parsing, converting unstructured log data into organized log templates. A static log parser is unsuitable for all logs due to the varying log data formats across different systems [11]. Log parsers may result in semantic loss of information, and their robustness and correctness greatly influence the efficacy of log anomaly detection.

Machine learning has used adaptive techniques for threat detection, employing algorithms that

analyze previous data to recognize trends and abnormalities. Methods, include decision trees, support vector machines (SVM), and neural networks, have been utilized to enhance detection accuracy and adaptability. Although these supervised learning models achieve high accuracy and flexibility, they require labeled datasets, limiting their application.

In recent years, the fast development of deep learning techniques has created new chances to tackle anomaly detection difficulties [12].Deep learning is a machine learning approach utilizing artificial neural networks to extract important features from extensive and complicated datasets, hence enhancing the detection of anomalous behavior [13].

In the domain of cybersecurity logs, researchers have started to adopt deep learning methods to enhance the effectiveness of anomaly detection and log analysis. These methods include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and more techniques, which have achieved significant results in identifying abnormal behavior [14].

However, despite the potential of deep learning methodologies for anomaly detection in time series analysis, they face several limitations. These limitations include handling high-dimensional data, interference from noise, data imbalance, and conceptual drift [15].

In light of the above, we propose a novel semi-supervised system for building a system with effective anomaly detection capabilities in system logs, enhancing cybersecurity applications. The framework consists of several layers that utilize semantic and statistical for hybrid feature representation, deep learning, traditional models, and ensemble techniques for anomaly detection. This multi-layer architecture ensures robustness, scalability, and adaptability in dynamic log anomaly detection. The main contributions of this work can be summarized as follows:

1) We presented a feature engineering technique that employs a hybrid feature representation, integrating lexical features (TF-IDF), semantic features (SBERT), and statistical features into each log entry, offering a full depiction of system logs.
2) Develop a Semi-Supervised Anomaly Detection Framework that integrates both labelled and unlabeled data by utilizing deep learning models (Auto-encoder, Bi-LSTM) alongside classical models (Isolation Forest, Single-Class SVM) to address the scarcity of labeled data in the cybersecurity field.
3) Designing a two-tier ensemble framework utilizing a weighted average (soft voting) clustering approach, succeeded by stacking with a Random Forest model as a meta-learner, to enhance the accuracy and reliability of anomaly detection in system logs.

The remaining parts of this work are structured as follows: Section 2 presents the related work. Section 3 delineates the proposed methodology for analyzing anomalous behavior with machine learning techniques. Section 4 delineates the experimental results and settings. Ultimately, Section 5 concludes and provides a future work.

2 RELATED WORKS

This section discusses related study in log-based anomaly detection, focusing deep learning methodologies.

In 2023, Y. Alaca et al. [4] presented Graph-Based Long Short-Term Memory (GLSTM), a graph-oriented deep learning model for cyberattack detection using log data. The system consolidates complex log data, trains an AI model on it, and finds anomalies. Node2Vec converts complex log data into graph data. The graph data is trained using LSTM, Bi-LSTM, and GRU deep learning algorithms. The proposed system was tested using HDFS, BGL, and IMDB datasets. Experimental findings showed that it performed best on the HDFS dataset, obtaining 97.01% accuracy. In 2023, Y. Lee et al. [16] presented LAnoBERT, a parser-free system log anomaly detection method using natural language processing's Bidirectional Encoder Representations from Transformers (BERT) methodology. LAnoBERT uses masked language modelling (MLM), a BERT-based pre-training strategy, and unsupervised learning for anomaly identification, using a loss function to model the masked language for each log key during assessment. The proposed LAnoBERT used masked language modelling to understand natural log data, and prediction error and predictive likelihood were used to identify anomalies during testing. To discover anomalies, they suggested scoring using top-k prediction probability. High-profile log datasets HDFS, BGL, and Thunderbird were tested. The HDFS F1 score was 0.9304. In 2023, Ch. Zhang et al. [17] introduced an approach for log sequence anomaly detection, termed LayerLog, which employs a three-layer "word-record-record sequence" structure of log data based on hierarchical semantics,, and uses Bi-LSTM models with attention

mechanisms. LayerLog operates without log parsers in the preprocessing step and can extract semantic characteristics from each layer. Experiments were performed on two widely utilized public datasets, HDFS and BGL, showing that the higher precision on the HDFS dataset was 0.996. In 2024, A. Aziz and K. Munir [12] introduced a hybrid anomaly detection method that integrates supervised and unsupervised learning. They employed self-organizing maps (SOMs), BERT encoders, and autoencoders. Utilizing the BERT encoder to generate semantic vectors and SOMs for the generation of clustering features. Autoencoders are employed for pattern recognition. The evaluation findings on two datasets: HDFS and BGL, indicate that the proposed technique attained an accuracy of 93% and a recall of 92%. In 2024, T. Rajendran et al. [18] integrated deep learning models with anomaly detection to improve the framework's ability to identify known and undiscovered threats. They used network traffic logs, historical incident records, and system logs for their investigation. Data is used to train deep learning methods like CNNs and RNNs. The suggested system trained and tested data in Jupyter Notebook. While SPSS is used for graphical prediction, G-Power is used for algorithm pretesting and computation to improve algorithm efficacy. Both deep learning approaches were evaluated using IBM SPSS. Iterations with 20 samples each were used to analyze and document accuracy. An RNN with 96% accuracy outperformed a CNN algorithm with 93% accuracy. In 2024, S. Wang et al. [19] introduced a hybrid model that synergistically blends Isolation Forest (IF), Generative Adversarial Network (GAN), and Transformer, with each component fulfilling a specific function. The Isolation Forest efficiently detects anomalous data points, the GAN generates synthetic data that mirrors the properties of the original data distribution to enhance the training dataset, and the Transformer is employed for modeling and context extraction in time series data. The experimental results were performed on four datasets: UNSW-NB15, NSL-KDD, CIC-IDS 2017, and Kyoto 2006+. The findings indicated that accuracy achieved was 94.67%.

3 PROPOSED METHODOLOGIES

This study presents a novel log anomaly detection system that utilizes a hybrid feature representation, statistical and semantic features, and integrates deep learning and traditional anomaly detection techniques. The framework comprises six sequential layers: the preprocessing and feature engineering layer, the preprocessing layer, the anomaly detection models layer, ensemble layer 1, ensemble layer 2, and the predictions layer. These layers work together efficiently to identify anomalies, enhancing cybersecurity. Figure 1 illustrates the configuration of the layers inside the proposed framework. The following is a description of each layer.

3.1 Preprocessing and Feature Engineering Layer

Data preprocessing is a crucial initial step applied to raw text data to prepare it for analysis. Analytical tools may yield wrong results and be misled if the data contains impurities, such as duplicates or missing data. Therefore, data preprocessing is required before conducting data analysis. We removed all records with missing and duplicated data [20].

Feature Engineering is an extraction of important features, including process ID, Date, Time, Level, Message, and Component. We applied the following algorithms to extract the features:

- TF-IDF (term frequency-inverse document frequency) Algorithm: It is for calculating the frequency weight of each word. This strategy relies on the notion that if a word frequently occurs in one log sequence and infrequently in others, it is more discriminative and highly significant. We can initially acquire TF and IDF values as follows [17]:

$$TF = \frac{\left|L^i\left(W_k^{ij}\right)\right|}{|S_i|} \quad . \tag{1}$$

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

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

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

$$FreqW_k^{ij} = \mathrm{TF} \times \mathrm{IDF}\,. \tag{3}$$

In our approach, each log entry is sparse represented by a vector of 1000 dimensions, mostly containing zero values:

- SBERT (Sentence BERT). Is a pre-trained neural network for text data to construct a semantic word vector. Semantic vectors are utilized to analyze log messages, enabling the comparison of patterns and similarities that identify anomalies. Implementing the SBERT Encoder enhances anomaly detection by effectively extracting contextual information from log messages, surpassing basic feature-based methods. The strategy bolsters credibility without compromising system breakdowns and cybercriminal operations. It facilitates the acquisition of essential embedding and simplifies log analysis through the capability to compare
- Using semantic data [12]. In our approach, each log entry is dense represented by a vector of 348 dimensions, all containing real values and not zeros as in TD-IDF.
- Statistical Features. We extracted statistical features, along with the prior feature extraction techniques, utilizing standard metrics to improve the system's performance. The metrics include text length, word count, vocabulary size, average word length, sentence length, punctuation frequency, capitalization ratio, and lexical diversity [21].
- Feature Concatenation: Concatenated the sparse, dense, and statistical features into an extensive matrix, with each entry encompassing all the above-mentioned features.

3.2 Preprocessing Layer

This layer enhances the data's readiness for the models employed in the subsequent layer. It minimizes computational complexity and focuses models to concentrate on critical information. It consists of the following steps:

1) Normalization (RobustScalar). It involves scaling characteristics utilizing statistics that are resilient to outliers. This Scaler eliminates the median and normalizes the data based on the quantile range, defaulting to the Interquartile Range (IQR). The interquartile range (IQR) is the difference between the first quartile (25th percentile) and the third quartile (75th percentile) [22].
2) Dimensionality Reduction (TruncatedSVD). This transformer executes linear dimensionality reduction by truncated singular value decomposition (SVD). This estimator, unlike PCA, does not center the data before performing the singular value decomposition. This indicates its capability to operate efficiently with sparse matrices [23].

3.3 Anomaly Detection Techniques layer

It is responsible for the initial detection of anomalies in the system using semi-supervised anomaly detection methods, including deep learning (supervised) and traditional (unsupervised) models. Before starting with these models, we split the dataset into 70% for training and validation, and 30% for testing. The models used in the framework are described below-

1) Deep Learning (Supervised) Models:
   - Auto-Encoder. It is a neural network model that compresses input data into a compact representation, which is subsequently decoded to reconstruct the original input. Auto encoders facilitate anomaly identification by assimilating a system's anticipated behavior and identifying deviations from it. It generate semantic vectors for log messages by integrating auto encoders, thereby maintaining the context and significance of the data [12].
   - LSTM. Long Short-Term Memory Networks (LSTM) can hold information for long periods due to their chain-like structure, where they can solve tasks that are difficult to implement using traditional RNNs. LSTM neural networks are structured for sequential data processing [4].
2) Traditional (Unsupervised) Models:
   - Isolation Forest. Is a tree-based unsupervised learning approach that employs the isolation of observations that are disparate from the rest of the input data. The method utilizes the creation of an ensemble of decision trees, each partitioning the data into smaller subsets [19].
   - One Class SVM. OCSVM is a prevalent one-class classification model employed for log anomaly detection, relying solely on normal log data. The model is engineered to delineate the boundary that distinguishes the bulk of input data from the residual, represented as a hyperplane that segregates normal data from outliers [18].
   - Extend IF. It is an enhancement of the traditional Isolation Forest algorithm. It was created to overcome some drawbacks of the original version, particularly in managing

high-dimensional data or where the distinctions between normal and anomalous data are intricate [19].

3.4 Ensemble Layer 1

This layer's primary goal is to aggregate the strengths of several models while diminishing the influence of weaker outcomes by decreasing their weights, yielding a balanced anomaly score that encapsulates the collective performance of all models. This layer first normalizes the data to ensure it ranges between 0 and 1. Subsequently, weights are determined for each model according to its performance assessed by the F2 metric, employed to identify potential cybersecurity threats. Finally, a weighted average of all outcomes is computed to represent the anomaly score.

3.5 Ensemble Layer 2

After merging the models into the previous layer, this layer improves the final decisions, where the decision threshold is tuned. This allows for determining the cut-off point that separates normal from abnormal to achieve the best balance between precision and recall. This is done using the Random Forest algorithm to capture the non-linear relationships between the model outputs. It takes the outputs of the previous models and re-trains itself on the labeled data to improve the final decision and generate a more reliable decision.

3.6 Predictions Layer

The final layer of the proposed framework converts the outcomes of the preceding ensemble learning layers into definitive conclusions that can be analyzed and utilized to improve cybersecurity. Each log is categorized as normal or abnormal, with a corresponding confidence score provided to the classification. The confidence score yields a probability value reflecting the degree to which the model suggests that the record constitutes an anomaly. This likelihood facilitates the categorization of occurrences by cybersecurity experts based on their significance.

4 EXPERIMENT RESULTS

4.1 Dataset

We evaluated the efficacy of our approach in identifying log anomalies within HDFS log datasets. This dataset was acquired from the extensive LogHub log collection, released by Jiming Zhu et al. [24]. The total count of log messages is 575,061, comprising 558223 normal logs and 16838 abnormalities. The training and validation dataset comprises 402,542 logs, whereas the test dataset comprises 172,519 logs.

4.2 Implementation Details

This work was executed using Python 3.13 and TensorFlow 1.13 on a Linux server equipped with an Intel(R) Core(TM) i7-8850H CPU @2.60 GHz and 32 GB of RAM.

4.3 Evaluation Metrics

To evaluate the effectiveness of the proposed anomaly detection framework, several standard performance metrics were employed, including accuracy, precision, recall, F1-score, and receiver operating characteristic (ROC) analysis. These metrics provide a comprehensive assessment of the model’s capability to correctly distinguish between normal and anomalous log events.

Accuracy measures the overall proportion of correctly classified log records, while precision evaluates the reliability of anomaly predictions by measuring how many detected anomalies are truly anomalous. Recall reflects the system’s ability to identify actual anomalous events within the dataset. The F1-score provides a balanced evaluation between precision and recall, particularly important in imbalanced datasets such as HDFS logs, where abnormal events are significantly fewer than normal events.

In addition, ROC-based analysis was utilized to examine the trade-off between detection sensitivity and false alarm rates. The true positive rate (TPR) represents the proportion of correctly detected anomalies, whereas the true negative rate (TNR) reflects the capability of the model to correctly identify normal log entries.

4.4 Result Analysis

The performance outcomes derived from the proposed models are presented in Table 1. Outcomes differ based on the model. In experiments performed on the HDFS dataset, the LSTM technique attained the maximum accuracy rate of 0.9996. The autoencoder achieved an accuracy of 0.9703. The OCSVM approach demonstrated a performance accuracy of 0.5923. The expanded IF technique demonstrated a performance accuracy of 0.5553. The IF technique demonstrated performance with an accuracy of 0.5365.

The outcomes of the prior models exhibited varying performance; hence, we employed an ensemble mechanism comprising two critical layers. The initial layer presents the outcomes of the model parameters. These weights denote the contribution of each model to the outcome following integration. Figure 2 illustrates the weight assigned to each model in the ensemble process. The weight computation employed F2-based weighting, resulting in an improved equilibrium between recall and precision in the models. The above figure indicates that the LSTM deep model is the predominant contributor to the ensemble outcomes, attributable to its capacity for temporal sequencing of the recordings. Conversely, the traditional models (IF and Extend IF) have facilitated advancements in anomaly identification, hence augmenting the system's performance cohesively.

Figure 3 illustrates the second-layer threshold optimization process, a key step in transforming values into final decisions (normal and abnormal). Values greater than the threshold are considered anomalous, while values smaller than the threshold are considered normal. The optimal threshold value (0.754) was determined using the (F2 Score) metric, which prioritizes recall. This threshold clearly separates normal from abnormal records, demonstrating the effectiveness of the ensemble system in improving final decisions and reducing errors.

Table 1: Results of the proposed models' performance.

Model	Accuracy	Precision	Recall	F1	F2
Isolation Forest	0.5365	0.0657	1.0000	0.1233	0.2601
Extend IF	0.5553	0.0683	1.0000	0.1278	0.2681
OCSVM	0.5923	0.0281	0.3427	0.0519	0.1058
Auto-Encoder	0.9703	0.5447	0.5414	0.5430	0.5420
LSTM	0.9996	0.9935	0.9947	0.9941	0.9944

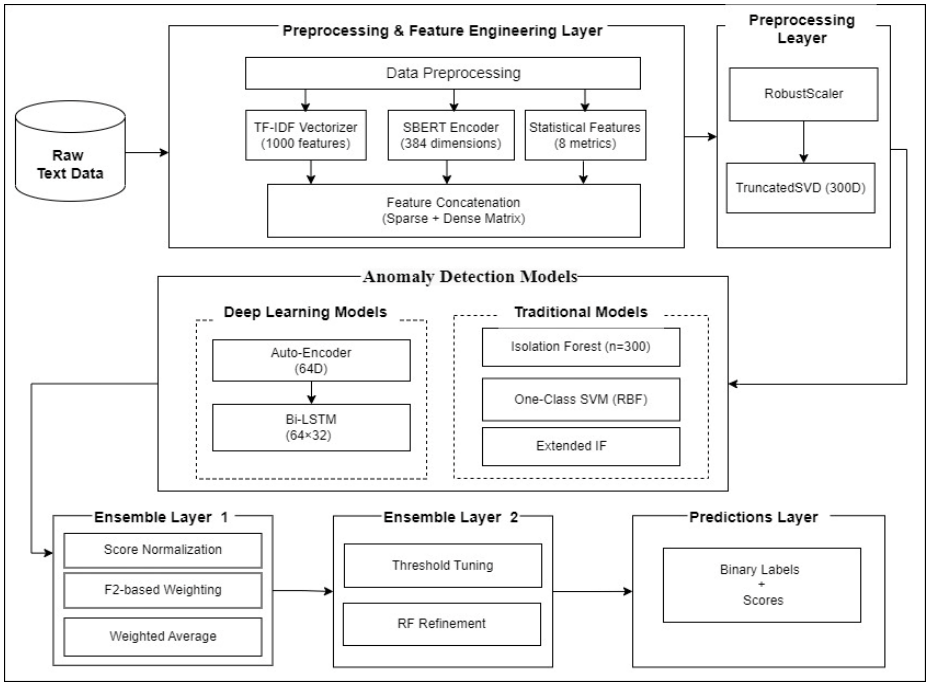

Figure 1: The framework of our work.

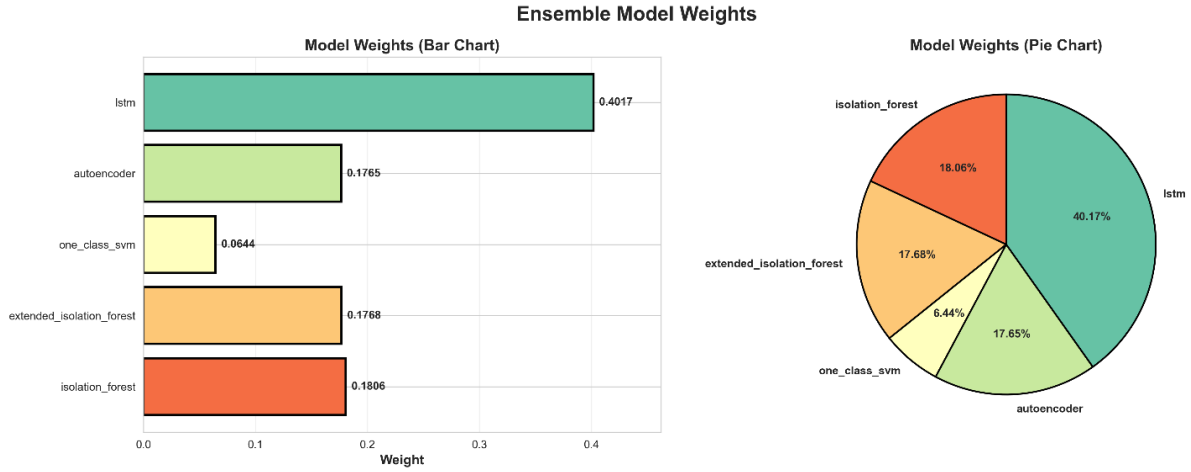

Figure 2: Ensemble model weights.

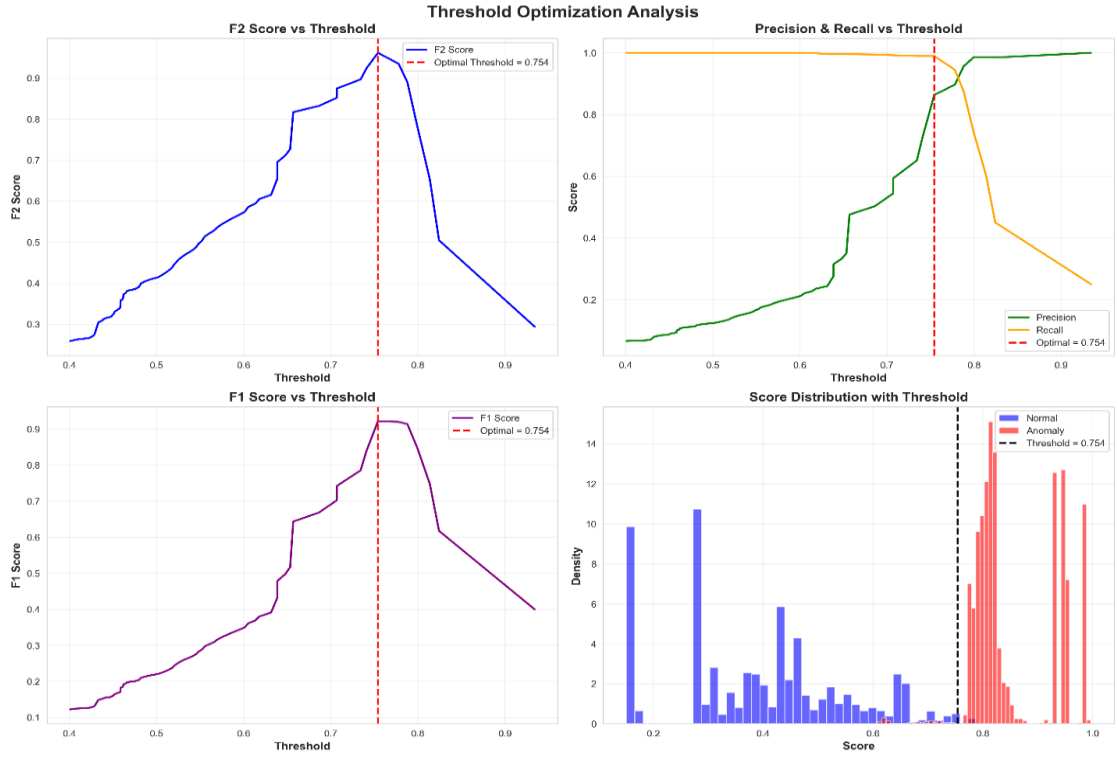

Figure 3: Threshold optimization analysis.

Figure 4 presents the confusion matrix, illustrating the success outcomes resulting from the tests conducted in this study. This graph assesses the efficacy of the actual versus predicted values. The crucial aspect is that the estimated values derived from training our model were juxtaposed with the real values, and their correctness was assessed. This graph illustrates numerous anomalies, with the real abnormality identified after the model's training. Consequently, the graph indicates that our model has attained significant success.

Two significant ratios are computed in the ROC-AUC curve. The true positive ratio is represented in (8). The alternative is the true negative ratio, as depicted in (9). Figure 5 illustrates the graph of the AUC curve. Lower values on the x-axis of the graph signify diminished false positives and elevated true negatives. The y-axis of the graph indicates elevated

values, signifying increased true positives and diminished false negatives. The suggested model achieves an AUC value 1.0000, signifying a perfect classification threshold between normal and abnormal situations. An AUC of 1.0 indicates that the model possesses exceptional discriminatory power with nearly no misclassifications.

The Precision-Accuracy graph is another metric that accurately evaluates the model. These curves are referred to as Sensitive Recall Curves. The precision indicates the accuracy of the model's positive predictions, as demonstrated in (4). The precision is demonstrated in (5). This facilitates a more precise assessment of true positives.

Figure 6 illustrates the graph of Precision against Accuracy. The integral of the area beneath the curve demonstrates the precision and efficacy of the model.

The proposed approach exhibited enhanced accuracy relative to previous experiments, as shown in Table 2.

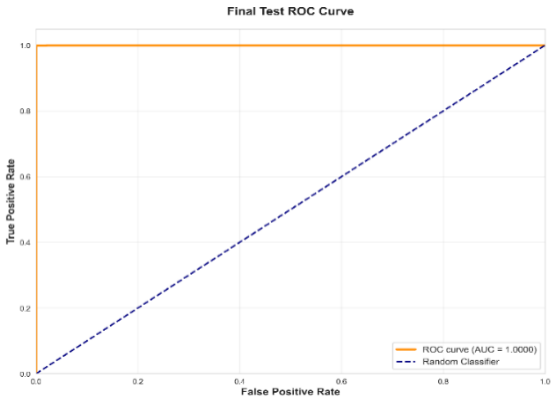

Figure 4: The confusion matrix of the test results of the suggested model.

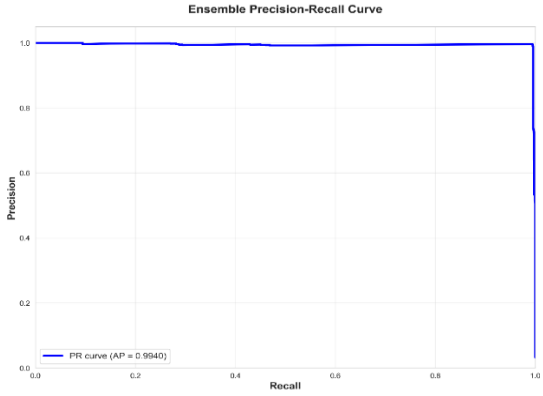

Figure 5: Test ROC curve.

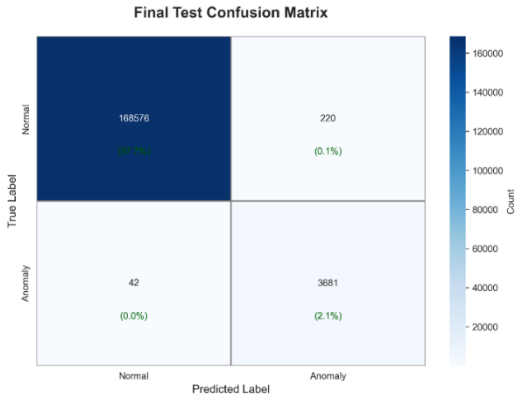

Figure 6: Precision-recall curve.

The above table shows the efficacy of machine learning and deep learning algorithms from prior studies with the suggested methodology. This comparison relies on the outcomes derived from each method. This research seeks to identify the optimal strategy for assessing anomaly detection in system logs that produces the most precise classification results in differentiating between normal and abnormal logs.

Table 2: A comparative analysis of the proposed methodology and prior studies utilizing the HDFS dataset.

Ref. and Year	Methodology	Result value
[4], 2023	LSTM, Bi-LSTM, GRU	Accuracy 97.01%
[16], 2023	BERT, MLM	F1 93.04%
[17],2023	Bi-LSTM	Precision 0.996
[12], 2024	SOMs, BERT, autoencoders	Accuracy 93%
[19],2024	IF, GAN, and Transformer	Accuracy 94.67
Our Proposed System	Auto encoder, LSTM, IF, Extend IF, and OCSVM	Accuracy=99.85% Precision=94.36% Recall= 98.87% F1=96.56% F2=97.94%

5 CONCLUSIONS

This paper proposes a novel semi-supervised ensemble framework for anomaly detection in cybersecurity logs. The semi-supervised learning approach markedly reduces reliance on fully labeled datasets, making the methodology more applicable to real-world cybersecurity contexts where labeled anomalies are scarce. This work uses a hybrid feature representation (sparse, semantic, and statistical) to improve anomaly detection accuracy. Based on deep learning and traditional models, anomalies in log data are detected. A two-layer ensemble technique is designed. The first layer relies on a weighted average to integrate the initial results, while the second layer uses a final filter model to improve decision accuracy and reduce false alarms. The suggested two-layer ensemble design offers significant scalability and flexibility, facilitating effortless adaptation to diverse system architectures and changing threat environments.

The findings validate the efficacy of the proposed methodology in analyzing system logs, demonstrating a detection accuracy surpassing 99% and attaining a ROC value. These results validate the proposed framework's efficacy and dependability, affirming the approach's feasibility. In the future, real-time log monitoring may be enhanced by integrating several anomaly detection models, including clustering, boosting, and cascade approaches. Moreover, integrating domain-specific knowledge and insights might augment the model's capacity to discern significant patterns and abnormalities.

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, vol. 3097, no. 1, 2024.

[2] S. A. H. Sándor R. Répás, "Anomaly Detection in Log Files Based on Machine Learning Techniques," J. Electr. Syst., vol. 20, no. 3s, pp. 1299-1311, 2024, doi: 10.52783/jes.1505.

[3] Y. Zhang et al., "Deep Learning for Anomaly Detection in Cybersecurity," ACM Trans. Cybersecurity, no. February, 2021.

[4] Y. Alaca, Y. Çelik, and S. Goel, "Anomaly Detection in Cyber Security with Graph-Based LSTM in Log Analysis," Chaos Theory Appl., pp. 188-197, 2023, doi: 10.51537/chaos.1348302.

[5] Y. Zhang, X. Chang, L. Fang, and Y. Lu, "Deeplog: Deep-learning-based log recommendation," in 2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2023, pp. 88-92.

[6] F. Hadadi, J. H. Dawes, D. Shin, D. Bianculli, and L. Briand, "Systematic evaluation of deep learning models for log-based failure prediction," Empir. Softw. Eng., vol. 29, no. 5, p. 105, 2024.

[7] D. S. M. Meena Siwach, "Anomaly detection for web log data analysis: A review," J. Algebr. Stat., vol. 13, no. 1, pp. 129-148, 2022.

[8] L. Yang et al., "Try with simpler-an evaluation of improved principal component analysis in log-based anomaly detection," ACM Trans. Softw. Eng. Methodol., vol. 33, no. 5, pp. 1-27, 2024.

[9] C. Egersdoerfer, D. Zhang, and D. Dai, "Clusterlog: Clustering logs for effective log-based anomaly detection," in 2022 IEEE/ACM 12th Workshop on Fault Tolerance for HPC at eXtreme Scale (FTXS), 2022, pp. 1-10.

[10] P. Jia, S. Cai, B. C. Ooi, P. Wang, and Y. Xiong, "Robust and transferable log-based anomaly detection," Proc. ACM Manag. Data, vol. 1, no. 1, pp. 1-26, 2023.

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

[12] 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.

[13] 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.

[14] M. Siwach and S. Mann, "Anomaly Detection for Web Log based Data: A Survey," in 2022 IEEE Delhi Sect. Conf. (DELCON), vol. 13, no. 1, pp. 129-148, 2022, doi: 10.1109/DELCON54057.2022.9753130.

[15] M. Fahim and A. Sillitti, "Anomaly Detection, Analysis and Prediction Techniques in IoT Environment: A Systematic Literature Review," IEEE Access, vol. 7, pp. 81664-81681, 2019, doi: 10.1109/ACCESS.2019.2921912.

[16] Y. Lee, J. Kim, and P. Kang, "LAnoBERT: System log anomaly detection based on BERT masked language model," Appl. Soft Comput., vol. 146, 2023, doi: 10.1016/j.asoc.2023.110689.

[17] 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.

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

[19] S. Wang, R. Jiang, Z. Wang, and Y. Zhou, "Deep Learning-based Anomaly Detection and Log Analysis for Computer Networks," J. Inf. Comput., vol. 2024, no. 2, pp. 34-63, 2024, [Online]. Available: https://doi.org/10.30211/JIC.202402.005.

[20] V. Çetin and O. Yıldız, "A comprehensive review on data preprocessing techniques in data analysis," Pamukkale Üniversitesi Mühendislik Bilim. Derg., vol. 28, no. 2, pp. 299-312, 2022.

[21] A. Sharma, M. Agrawal, S. D. Roy, V. Gupta, P. Vashisht, and T. Sidhu, "Deep learning to diagnose Peripapillary Atrophy in retinal images along with statistical features," Biomed. Signal Process. Control, vol. 64, p. 102254, 2021.

[22] M. M. Lasiyono, N. Nurhayati, T. G. Soares, and M. Mulyadi, "Enhancing Support Vector Machine Performance for Heart Attack Prediction using RobustScaler-Based Outlier Handling," Bull. Informatics Data Sci., vol. 4, no. 1, pp. 1-9, 2025.

[23] 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.

[24] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," in Proc. ACM SIGOPS 22nd Symp. Operating Systems Principles, 2009, pp. 117-132.