MINIMALISTIC FAULT DETECTION IN SYSTEM LOGS USING UNSUPERVISED LEARNING

BY

MD. IFTEKHARUL ISLAM RIDOY
ID: 242-25-038

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Computer Science and Engineering

Supervised By

Md. Abbas Ali Khan
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised By

Abdus Sattar
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

13 SEPTEMBER, 2025

APPROVAL

This Thesis, titled **"Minimalistic Fault Detection in System Logs Using Unsupervised Learning"**, submitted by **Md. Iftekharul Islam Ridoy**, ID No: **242-25-038** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 13-09-2025.

BOARD OF EXAMINERS

Chairman

Dr. Arif Mahmud
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Ms. Nazmun Nessa Moon
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Md Alamgir Kabir
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner

Mr. Sadat Hossain
Data Scientist
Risk Management Division,
BRAC Bank Limited

DECLARATION

I hereby declare that, this research has been done by me under the supervision of **Md. Abbas Ali Khan, Assistant Professor, Department of CSE** Daffodil International University. I also declare that neither this research nor any part of this research has been submitted elsewhere for award of any degree or diploma.

Supervised by:

27.08.25

Md. Abbas Ali Khan
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Abdus Sattar
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Md. Iftekharul Islam Ridoy
ID: 242-25-038
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, I express my heartfelt thanks and gratitude to Almighty God for His divine blessing, which made it possible for me to complete the final year of research successfully.

I am really grateful and wish to express my profound indebtedness to **Md. Abbas Ali Khan**, **Assistant Professor**, Department of CSE, Daffodil International University, Dhaka, has deep knowledge & keen interest in the field of Machine Learning to carry out this research. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this research.

I would like to express my heartiest gratitude to the Head**,** Department of CSE, for his kind help to finish my research and also to other faculty member and the staff of CSE department of Daffodil International University.

Finally, I must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

In a computing system, the system logs are important for detecting the issues and failures and anomaly. The modern systems are producing massive amount of log daily. The traditional detecting methods are usually unable to detect complex issues. This study aims for a machine learning based method that will detect anomalies in the system logs. Log data is the first transformed into structured templates and represented them using TF-IDF method. The anomaly detectiion models, with Isolation Forest and Local Outlier Factor (LOF), are then applied to detect issues and failures. The Experiments shows that the LOF (Local Outlier Factor) achieves higher accuracy and recall compared to IF (Isolation Forest), which is showing the strong potential for practical assessment. This study results highlight that with the combination of feature engineering along with lightweight machine learning methods enables efficient and automated log anomaly detection, and improving system reliability with reducing manual monitoring efforts.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Introduction

The computer systems produces massive amount of logs daily that records about the information of the system operations and application behavior, as well as the failures. Those logs are the source for monitoring and detecting system health. This is difficult to identify the failures and issues in log data for their unstructured log representation and behaviour. As the result, automated detecting techniques for anomaly detection in system logs are becoming increasingly important for ensuring reliability.

The previous methods are mainly uses manual rule-based monitoring or regular expressions are insufficient to handle the variability, along with evolving patterns of system logs. These techniques can efficiently capture known issues and failures, but they often suffer from limited adaptability and struggle to identify previously anomalies which are not seen. And, the manual crafting and maintaining rules are requiring significant domain expertise.

I used the application of unsupervised learning in this study for system log anomaly identification and employ template feature extraction using TF-IDF to represent log templates as numeric values with capturing their importance within the dataset. This features are then utilized by using the selected models which are the Isolation Forest and the Local Outlier Factor (LOF).

The proposed methodology is utilized on HDFS_2k log dataset collected from open source, and the performance measured using metrics are - Recall Accuracy, Precision, F1-score, and ROC-AUC. The experiments finds the highlight that Isolation Forest combined with TF-IDF template extraction outperforms LOF, based on the accuracy as well as the recall. The results demonstrates about the uniqueness of integrating template feature engineering with unsupervised learning for system log anomaly detection.

1.2 Motivation

Digital systems are producing a lot of logs which is massive, and with these, the previous analog methods are not so practical at all for monitoring. The previous analog rule based methods usually fail to identify the previously unseen anomalies and leaving systems vulnerable leads to system failures and security threats and cause of cyber theat. This project harnesses ML models to automatically detect along with identify anomalous behaviour patterns in system logs data, enabling faster incident detection, reducing operational risks, and enhancing overall system reliability. By adopting such intelligent approaches, organizations can shift from reactive troubleshooting to proactive system monitoring.

1.3 Rationale of the Study

This study addresses the gap between sophisticated deep learning models and practical, lightweight solutions. By focusing on unsupervised methods with TF-IDF feature extraction, we demonstrate that competitive anomaly detection can be achieved without relying on expensive infrastructure or template-based preprocessing. The rationale is that even simple statistical embeddings and classical machine learning models can uncover abnormal system behavior effectively, provided they are designed thoughtfully.

1.4 Research Questions

i. How effectively can machine learning algorithms detect anomalies in system logs compared to traditional methods?
ii. Which log data features or representations best improve anomaly detection accuracy?
iii. How do models like Isolation Forest and Local Outlier Factor compare in performance and efficiency?
iv. Can automated anomaly detection enhance system reliability and reduce operational risks in practice?

1.5 Expected Output

The study is expected to produce a machine learning-based model capable of accurately detecting anomalies in system logs, identify the most impactful log features, compare the performance of different algorithms, and demonstrate a lightweight solution for near real-time monitoring to improve system reliability and reduce operational risks.

1.6 Project Management and Finance

Project Management:

The project followed a structured approach divided into key phases:

- Planning: Objectives, scope, research questions, and required resources were defined.
- Data Collection and Preprocessing: System log datasets were acquired and prepared for analysis.
- Model Development: Isolation Forest together with Local Outlier Factor were implemented and trained for anomaly detection.
- Evaluation and Validation: For model performance used the metrics like the accuracy, the F1-score along with the precision and the recall, as well as the ROC-AUC.
- Documentation and Reporting: Findings, challenges, and practical recommendations were summarized.

Project management involved task scheduling, progress tracking, and milestone reviews, which ensured timely completion and quality outcomes.

Finance:

The financial requirements of the project were minimal, as it primarily relied on publicly available log datasets and open-source software libraries (Python, Scikit-learn, Pandas). Major expenses included:

- Computing Resources: Cloud or local servers were used for model training and testing.
- Software and Tools: Optional subscription fees were considered for advanced tools, if required.
- Miscellaneous Expenses: Minor costs were incurred for documentation, printing, and administrative needs.

Overall, the project was cost-effective, leveraging freely available resources and lightweight algorithms to achieve the research objectives.

1.7 Report Layout

This study has six chapters:

- **Chapter 01: Introduction**

  An overview of the thesis is given with the research problem, motivation, rationale, research questions, as well as expected outcomes.

- **Chapter 02: Background**

  Reviews the related works with key terminologies, comparative analysis with this each other, and scoope of the problem, and challenges.

- **Chapter 03: Research Methodology**

  Describes the dataset, data collection and the proposed methodology.

- **Chapter 04: Results and Discussion**

  Explains the experimental-Setup, presents the results with analysis, along with discusses findings.

- **Chapter 05: Impact on Society, Environment, and Sustainability**

  Examines societal, environmental, ethical, and sustainability aspects.

- **Chapter 06: Summary and Future Work**

  Summarizes the study, provides conclusions, offers recommendations, and the future research work.

CHAPTER 2
BACKGROUND

2.1 Preliminaries and Terminologies

Before implementing the anomaly detection system, several fundamental concepts and terminologies were defined and clarified:

- **System Logs:** Records generated by operating systems, applications, and network devices that captured events, errors, or system behavior over time. These logs were essential for monitoring, troubleshooting, and security analysis.
- **Anomalies:** Patterns in log data that deviated from expected behavior. Anomalies included rare events, errors, or unusual sequences that could indicate system failures, security breaches, or performance issues.
- **Feature Extraction:** This technique used for representing the system logs in a numerical vectorized form which is suitable for unsupervised machine learning algorithms.
- **Log Parsing:** Convert the systems raw logs messages into structured formats, which are the templates, to utilize the automated analysis and predict.
- **Machine Learning Models:**
    - **Isolation Forest:** This is a tree-based model which isolated the anomalies by recursively partitioning data points of the logs data.
    - **LocalOutlier Factor:** This is a density based algorithm which is responsible for identifying the outliers of the logs by comparing local density differences among neighboring data points of the logs.
- **Evaluation Metrics:** Used the evaluation metrics which are the F1-score along with the accuracy, the precision, and the recall, as well as the ROC-AUC to evaluate model performance and effectiveness in identifieing anomalies.
- **Unsupervised Learning:** These methods were applied to point out the patterns of the behaviour and the outliers.

2.2 Related Works

The previous approaches primarily used log parsing and rule-based systems, where handcrafted templates or statistical thresholds were used to idenify values. The researchers[1] introduced a deep LSTM-based methods that detect the next log sequence to predict anomalies of the systems logs.

The researcher [1][2] used unsupervised machine learning to reduce dependency on labeled data and demonstrated that deep learning could capture sequential dependencies in logs, albeit with high complexity. In contrast, the reasearchers[3] showed that clustering based methodology that could achieve competitive anomaly detection performance with lower computational demands.

The recent study, the researchers [4] compared classical and modern techniques and showing that simpler methods like Isolation Forest [5] and One-Class SVM [6] performed well when coupled with effective log representations. This key finding aligned with the researchers[5], which highlighted the efficiency of isolation-based approaches for high-dimensional anomaly detection.

Deep learning methods are the resource heavy as well as slow also, and recent studies emphasized the need for lightweight, template-free methodology that balanced performance and scalability. This study demonstrated that minimalistic worlkflow that could provide reliable anomaly detection with reducing computational cost with the use of TF-IDF vectorization and the Isolation Forest and the One-Class SVM.

2.3 Comparative Analysis and Summary

Table 2.1: Comparative Analysis

Aspect	Related Works (Papers)	My Work
Models Used	Deep learning methods and IF+LOF (hybridly)	Simple machine learning (IF & LOF)
Data Preprocessing	Often raw logs and embeddings	Template Extraction utilizing TF-IDF
Complexity	High	Low
Focus	Accuracy and advanced architectures	Efficiency and interpretability

It is evident that no single method offered a complete solution to identification of the anomalies in the system logs. Deep-learning models provided the sequence modeling at the expense of efficiency, where the unsupervised machine learing excelled in scalability but often suffered from lower precision due to the imbalanced datasets.

This study addressed this solution by implementing a lightweight anomaly detection method using TF-IDF vectorization with combining with Isolation-forest and Local-Outlier Factor. These approaches leveraged the strengths of unsupervised learning with ensuring computational efficeiency, and aligning with the recent research toward scalable, template-free, as well as resource-efficient anomaly detection models.

2.4 Scope of the Problem

This study focused on the problem of identifying the anomalies in large-scale system logs, which were often unstructured, high-dimensional, and highly imbalanced. The scope was limited to applying unsupervised machine learning techniques, specifically Isolation Forest and LOF, combined with TF-IDF-based log representations. Unlike deep learning methods, which required extensive labeled datasets and heavy computational resources,

the study emphasized lightweight and then scalable-models suitable for near real-time anomaly detection.

The research was bounded to the analysis of publicly available datasets such as HDFS_2k.log, where log parsing, feature extraction, model training, and evaluation were performed. The scope did not include the deployment of the system in production environments or integration with large-scale distributed monitoring platforms, but it laid the groundwork for such applications.

2.5 Challenges

During the study, several challenges were encountered:

- **Data Imbalance:** Anomalies were rare compared to normal events, making it difficult for models to learn meaningful boundaries and often leading to high false negatives.
- **Unstructured Nature of Logs:** System logs lacked a standardized format, requiring preprocessing and template generation (via parsing and TF-IDF) before being fed into machine learning models.
- **Feature Representation:** Capturing meaningful patterns from raw logs was non-trivial, as overly simplistic features risked losing contextual information, while complex representations increased computational costs.
- **Model Selection:** Choosing models that balanced accuracy, precision, and computational efficiency was challenging, especially when comparing lightweight unsupervised methods with resource-heavy deep learning approaches.
- **Evaluation-Metrics:** Standard metrics: “accuracy” were often misleading in highly imbalanced datasets, requiring careful consideration of recall, precision, F1-score, and ROC-AUC to fairly evaluate the experiment performance.
- **Generalization:** Models trained on a specific dataset (e.g., HDFS logs) risked overfitting and might not generalize well to logs from different systems or applications.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Proposed Methodology

The proposed methodology of this study was designed for identify as well as detecting the anomalies in system logs using a lightweight machine learning pipeline. The approach focused on balancing **accuracy** as well as the **computational efficiency**, avoiding the heavy overhead of deep learning models.

The methodology consisted of four primary phases: **log preprocessing**, **feature extraction**, **anomaly detection modeling**, and **performance evaluation**.

The provided diagram 3.1 outlines a structured methodology for log preprocessing, feature extraction, anomaly detection modeling, and performance evaluation

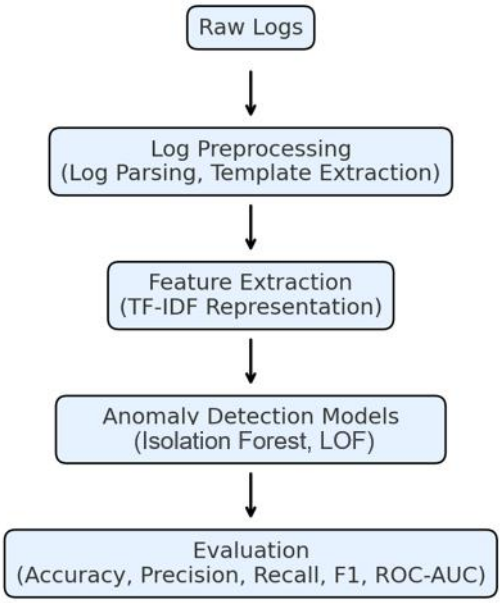

Figure 3.1: The working process to detect HDFS Log Anomaly

3.1.1 Log Preprocessing

Raw logs generated by distributed systems are typically unstructured and noisy. To make them suitable for machine learning, they were first parsed into structured **log templates**. Parsing grouped log messages with similar event patterns, removing parameters such as timestamps or variable identifiers that do not carry anomaly-related semantics.
For instance, a raw log entry such as:

blk_12345 failed to be replicated to node3

was parsed into the template:

blk_* failed to be replicated to node*

This step reduced dimensionality and standardized the input for subsequent feature extraction.

3.1.2 Feature Extraction Using TF-IDF

Since log templates are textual data, the logs were transformed into the numeric vectors by using Term-Frequency(TF) as well as Inverse-Document Frequency(IDF).. TF-IDF is specialized in capturing the importance of terms in textual data while down-weighting common but less informative words.

For a given term "*t"* in document "*d*" from a - corpus *D* = the TF-IDF weight was denotedas:

$$\mathrm{TF-IDF(t, d, D)} = \mathrm{TF(t, d)} \times \mathrm{IDF(t, D)}$$

Where:

- ***Term-Frequency(TF):***

$$\mathrm{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

With ft, d , = in a document = d, defines the frequency of term = t.

- ***Inverse-Document-Frequency(IDF):***

$$\mathrm{IDF}(t, D) = \log \quad \frac{|D|}{1 + |\{d \in D : t \in d\}|}$$

This transformation produced a sparse, high-dimensional feature space where rare but discriminative log patterns were emphasized, improving anomaly detection sensitivity.

3.1.3 Anomaly Detection Models

Two unsupervised models were employed: Isolation-Forest (iForest) and Local-Outlier Factor (LOF). Both of the algorithms were choosed because of their lightweight, effective for high-dimensional data, and do not require extensive labeled datasets.

a) Isolation Forest

Isolation Forest relies on the premise that anomalies are "few and different" and can be easily isolated using recursive partitioning. An algorithm picks a random feature, splits the data into two groups with values above and below some midpoint, and selects a value on the line between them. Anomalies, being rare, are isolated in fewer splits and therefore yield shorter path lengths.

A data point x's anomaly score can be found using:

$$s(x, n) = 2^{-\frac{E[h(x)]}{c(n)}}$$

Where:

- $\mathrm{h(x)}$ = defines the path length of xxx in the trees,
- $\mathrm{E\big(h(x)\big)}$ = defines the expected path length,
- $\mathrm{c(n)}$ = the averge path length in a binary search tree with nnn samples.

A score bear to = "1" indicates high anomaly likelihood, and the scores closer to = "0" suggest to normality.

b) Local Outlier Factor (LOF)

This is a density based anomaly detection system which is responsible for identifying the anomalies by comparing the local density of each data-point with the densities of its

neighbors in log data.. A log is idenfied as anomalous when a point have the lower density other than the neihgbor,

Local Outlier Factor is particularly effective for identifying the anomalies in the datasets with local densities, such as system logs represented as TF-IDF templates.

3.1.4 Model Evaluation

IF and LOF were trained and tested over the log dataset. The anomaly labels provided by the dataset enabled the quantitative evaluation. The standard metrics were applied:

- ***Accuracy***

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- ***Precision***

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- ***Recall***

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- ***F1-Score***

$$\text{F1} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{F P} + \text{FN}}$$

- ***ROC-AUC:*** Difference between the true positve and the false positve were measured.

3.1.5 Summary of Methodology

The methodology combined with TF and IDF, feature extraction with unsupervised_ learning for detecting the anomalies in the system logs. LOF leveraged the local density to detect the anomalous sessions. Isolation Forest exploited of the feature space to isolate rare anomalies. Focusing on lightweight models the proposed approaches ensured computational efficiency,and suitable for real-time system monitoring.

3.2 Dataset Collection

The dataset ***HDFS_2k.log***, was choosed from the open sourse Hadoop-Distributd File System logs. These logs provided the detailed records of a system-level events, containg the timestamps, the severity levels, and the log message contents which is making them highly suitable for anomalies-detection.

Table 3.1, the tabular representation of the dataset(HDFS) raw logs, which contains the timestamp of each log entry, and its severity level, and the corresponding message contents.

Table 3.1: Sample Raw Log Entry of HDFS Dataset

Timestamp	**Severity**	**Message**
2010-02-23 10:00:01	INFO	Namenode started with version 0.20.2
2010-02-23 10:00:05	WARN	DataNode registration failed: Connection timeout
2010-02-23 10:00:10	ERROR	File /user/hadoop/file1.txt replication failed

Table 3.2 shows the preprocessed the dataset after session grouping and the template extraction. Each session has the multiple log entries into the log templates, the structural patterns of events removing variable components such as file names, IDs, or paths. This structured format is necessary for the feature extraction and anomaly detection.

Table 3.2: Sample session after preprocessing and template extraction

Session ID	Log Templates	Severity Sequence	Label
S0001	Namenode started; DataNode registration failed; File replication failed	INFO → WARN → ERROR	Anomalous
S0002	Namenode started; Task completed	INFO → INFO	Normal

In this structured format, we got:

- Each session is represented as a sequence of log templates and their associated metadata.
- Labels indicate whether the session is normal or anomalous, allowing supervised or semi-supervised anomaly detection.

3.3 Implementation Requirements

The proposed anomaly detection framework is designed to be lightweight and feasible on standard computing infrastructure, without requiring specialized hardware. The implementation relies on **Python 3.8+** as the primary programming language, chosen for its extensive libraries for data processing, machine learning, and visualization.

Software and Libraries:

- **scikit-learn:** For implementing unsupervised models, including Isolation Forest and LOF, and for computing metrics are the Precision along with the Recall as well as the F1-Score, as well as the ROC-AUC.
- **pandas & NumPy:** For efficient data manipulation, preprocessing, and feature matrix creation.
- **Matplotlib & Seaborn:** For the visualization of confusion matrices and performance comparisons.

- **re (Regular Expressions):** To parse and group logs by session identifiers or Block IDs.

Hardware Requirements:

- A standard workstation or laptop with at least 8 GB RAM and a modern CPU is sufficient for experiments.
- No GPU is required since the framework does not rely on deep neural networks, keeping memory and computational overhead minimal.

Data Requirements:

- Access to HDFS in CSV or raw log format.
- A labeled file (e.g., anomaly_label.csv) for evaluation purposes.

CHAPTER 4
EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Results & Analysis

The research is done to measure the effectiveness of unlike anomaly-detection techniques on the HDFS_2k dataset. The experiments focused on two widely used methods for log anomaly detection, which are Isolation-Forest (IF) along with the Local-Outlier-Factor (LOF), both applied on TF-IDF–based log template features. The analysis considers detection performance, model behavior, and the impact of the feature representation strategy.

4.1.1 Performance Metrics

In order to evaluate the effectiveness of the program, the following metrics were used:

1. **Precision**: An indicator of the proportion of detected anomalies that were true anomalies, which illustrates the impact of false positives.
2. **Accuracy**: Determines whether the model has correctly classified normal and anomalous sessions.
3. **Recall (Sansitivity)**: Represents the propportion of true anomalies that were accurately detected, critical for anomaly detection where missing an anomaly can be costly.
4. **F1-Score**: The mean of the precision and the recall along with providing a balanced measure between over- and under-detection.
5. **ROC-AUC**: Specialized to distinguised between normal session and anomalous sessions across varying thresholds.
6. **Confusion Matrix Components (TP, FP, TN, FN)**: Offers insight into misclassification patterns, particularly the alikeness between detecting true anomalies and false alarms.

Among the evaluated methods, the best overall performance was achieved by LOF with TF-IDF templates, which obtained the highest F1-score (0.1698), accuracy (95.59%), precision (0.2368), and recall (0.1324) among all tested models.

Table 4.1: Experimental Results of Anomaly Detection Models on the HDFS_2k Dataset

Method	Accuracy	Precision	Recall	F1-Score	ROC-AUC
LOF + TF -IDF Templates	0.9559	0.2368	0.1324	0.1698	0.5586
Isolation Forest + TF-IDF Templates	0.9433	0.1538	0.1471	0.1504	0.5593

Result:

- **Overall Accuracy:** LOF with TF-IDF templates achieved the highest accuracy of 95.59%, largely due to the dataset being heavily skewed toward normal sessions.
- **Recall Differences:** The best Isolation Forest configuration achieved a recall of 0.1471, compared to LOF's 0.1324, indicating that Isolation Forest detected slightly more true anomalies, though both models struggled due to class imbalance.
- **Precision Differences**: LOF's precision was 0.2368, higher than Isolation Forest (0.1538), suggesting that LOF's anomaly predictions were more reliable, with fewer false alarms relative to its detections.
- **F1-Score Analysis:** LOF's highest F1-score was 0.1698, slightly higher than Isolation Forest's 0.1504, and showing that LOF provides the best trade-off between detecting anomalies and minimizing false predictions among the tested configurations.
- **ROC-AUC Performance:** Local Outlier Factor is more slightly give the best result againset of Isolation Forest in terms of ROC-AUC values but with low recall and precision.

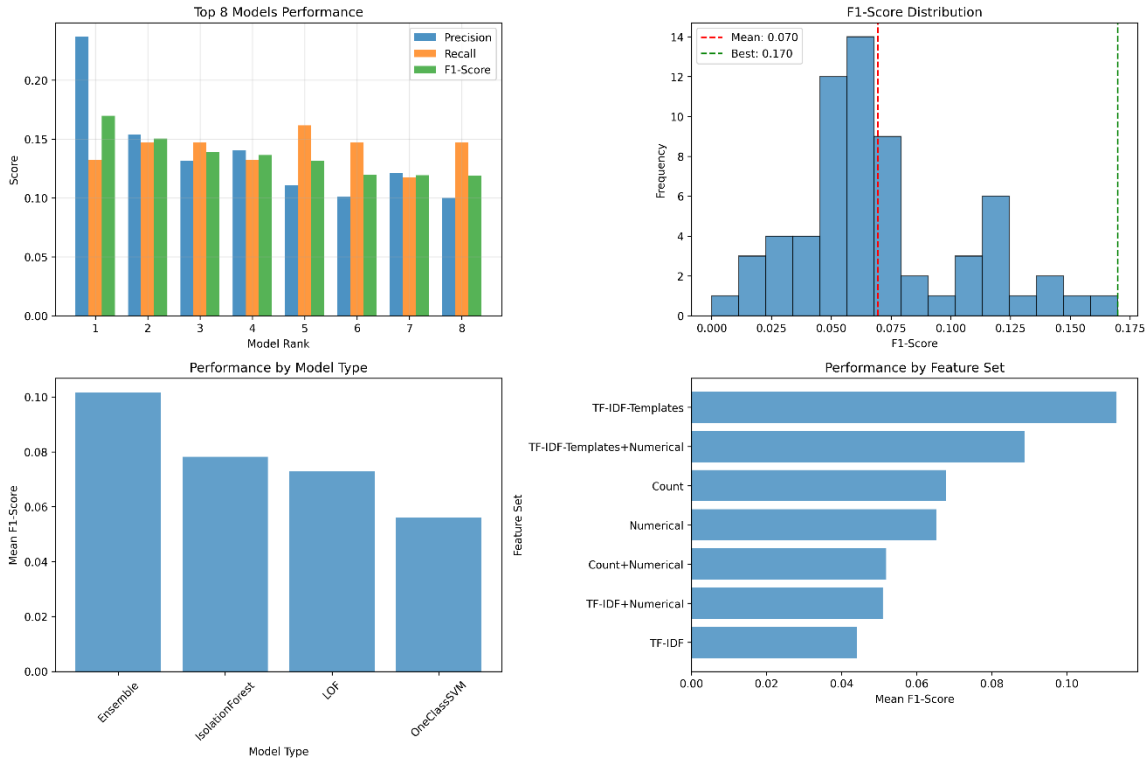

Figure 4.1: Performance Analysis

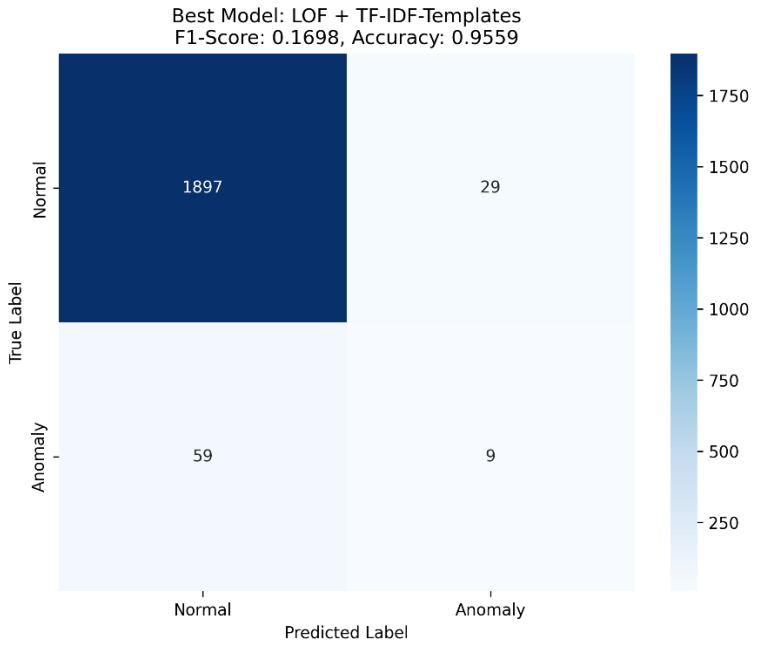

Figure 4.2: Best Confusion Matrix

4.1.3 Analysis of Model Behavior

- LOF's was the most effective in this experiment, achieving the highest of the F1-score and the precision among the tested configurations. This reliance on local neighborhood density allows it to identify structural values in TF-IDF template space better than Isolation Forest for this dataset.
- Isolation Forest isolates the anomalies using random feature partitioning, showed slightly higher recall but lower precision and F1-score. This shows that while it detects some anomalies missed by LOF, its detections are less reliable overall in this sparse TF-IDF feature space.
- Representing logs as TF-IDF template features helped to reduce dimensionality and emphasized repetitive log patterns.
- With only 3.41% of sessions being anomalous, dataset imbalance heavily affects detection metrics. Accuracy alone (>92% for all models) can be misleading; F1-score, precision, and recall provide a clearer picture of anomaly detection performance.

4.1.4 Summary of Findings

- LOF + TF-IDF templates is the top-performing model for this experiment, achieving the best balance between the precision as well as the F1-score.
- Isolation Forest shows slightly higher recall in some configurations but is less reliable overall, resulting in lower precision and F1-score.
- Feature representation is critical: TF-IDF templates provide a compact, informative encoding of log patterns, but future work could explore sequence-based embeddings or semantic representations to capture more subtle anomalies.
- Error analysis and improvement potential: Low recall across all models suggests that tuning contamination levels, applying adaptive thresholds, or combining multiple methods could improve detection coverage.

4.2 Discussion

The experiment results showed the strengths as well as the limitations of LOF and Isolation Forest in detecting anomalies in the HDFS_2k dataset. According to the latest ranking table, LOF with TF-IDF templates achieved the best performance from others, acquirng with the highest F1-score (0.1698), accuracy (95.59%), and precision (0.2368). Although Isolation Forest achieved slightly higher recall (0.1471 vs 0.1324), its lower precision and F1-score indicate that it produces more false positives relative to true anomalies.

The results underscore the impact of dataset imbalance: with only 3.41% of sessions being anomalous, metrics like recall and F1-score are more informative than accuracy.

In the conclusion, LOF with TF-IDF templates provides the best result for precision and F1-score for HDFS log anomaly detection in this method.

CHAPTER 5
IMPACT ON SOCIETY, ENVIRONMENT, AND SUSTAINABILITY

The proposed methodology is to improved reliability and security of a computing system, particularly in large distributed operating systems in example of the cloud services, data centers etc. By identifying the anomalies in system logs, the administrators can identify issues as well as the failures. This has a benefit, as many critical services (e.g., banking, healthcare, e-governance) depends on IT systems. The lightweight and unsupervised method reduces dependency on specialized expertise and expensive resources, making anomaly detection more accessible for small and medium-sized organizations. The lightweight model such as LOF with TF-IDF templates assures sustainability as well as the efficient model that makes easy resource consumption.

CHAPTER 6

SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH

6.1 Summary

This study utilized LOF and Isolation Forest for anomaly detection in the dataset using TF-IDF templates. Local outlier factor with TF-IDF templates bring off the best overall performance, along with the highest F1-score, the accuracy, and the precision, and the Isolation Forest showed slightly higher recall but lower overall reliability. This study puts an significance of feature representation and the challenge of identify and detecting the anomalies in a imbalanced dataset.

6.2 Further Study

1. Explore semantic embeddings, sequence-based representations, or graph-based features to detect more subtle or contextual anomalies.
2. Improve detection performance, balancing precision and recall more effectively in highly imbalanced datasets.

242-25-038

ORIGINALITY REPORT

19%	18%	5%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	9%
2	Submitted to Daffodil International University Student Paper	2%
3	Submitted to Higher Education Commission Pakistan Student Paper	2%
4	www.coursehero.com Internet Source	1%
5	Azeddine Mjahad, Alfredo Rosado-Muñoz. "Automated Quality Control of Candle Jars via Anomaly Detection Using OCSVM and CNN-Based Feature Extraction", Mathematics, 2025 Publication	1%
6	drpress.org Internet Source	<1%
7	bmcmedinformdecismak.biomedcentral.com Internet Source	<1%
8	actapress.com Internet Source	<1%
9	www.mdpi.com Internet Source	<1%
10	Submitted to Dublin Business School Student Paper	<1%
11	test-api.ijosi.org Internet Source	<1%

12	George M. Messinis, Nikos D. Hatziargyriou. "Unsupervised Classification for Non-Technical Loss Detection", 2018 Power Systems Computation Conference (PSCC), 2018 Publication	<1 %
13	Submitted to University of Sydney Student Paper	<1 %
14	peerj.com Internet Source	<1 %
15	www.ijsred.com Internet Source	<1 %
16	Aanonsen, Allan Karl. "Advanced Data Science Model for Detecting and Classifying IoT Malware", The University of Texas at San Antonio Publication	<1 %
17	dokumen.pub Internet Source	<1 %
18	www.frontiersin.org Internet Source	<1 %
19	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical and Computer Technologies", CRC Press, 2025 Publication	<1 %
20	www.ir.juit.ac.in:8080 Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		