LOG BASED TESTING THROUGH MACHINE LEARNING FOR HOSPITAL MANAGEMENT SYSTEMS

DR. ABDUL AZIZ
Department Head of Software Engineering
Fast NUCES Khi
Karachi,Pakistan
abdulaziz@nu.edu.pk

MARYAM RAHEEM
Department of Software Engineeing
Fast NUCES Khi
Karachi,Pakistan
k201700@nu.edu.pk

HAFSA BAIG
Department of Software Engineering
Fast NUCES Khi
Karachi,Pakistan
k20163@nu.edu.pk

***Abstract*—Our research project aims to explore the application of machine learning algorithms for log-based testing in hospital systems. We intend to investigate how machine learning can enhance data integrity and security through the detection of anomalies in system logs. This proposal provides an overview of the background, significance, methodology and expected outcomes of our research. This project delineates an advanced anomaly detection system for hospital environments, utilizing a decision tree algorithm to facilitate real-time data analysis and identify potential irregularities. Integrating a React.js frontend with a Node.js backend using the Express framework and MongoDB database, the system provides an intuitive interface for administrators to input and manage hospital data. Preprocessing techniques ensure data readiness for analysis, while anomalies are dynamically presented on the frontend via a table, complemented by a dashboard showcasing anomaly frequency and patient statistics. The system includes a notification alert mechanism, sending anomaly alerts via Gmail, thus enabling prompt rectification. Rigorous testing underscores the system's efficacy in enhancing hospital monitoring and management.**

Keywords— machine learning, testing, system logs, hospital management system, anomaly detection, decision trees.

I. INTRODUCTION

Hospital management systems (HMS) are vital for efficient healthcare operations and service delivery in today's evolving landscape. The objective of this research paper is to embark on a journey in log-based testing, for hospital management systems. This initiative utilizes state of the art machine learning algorithms to identify irregularities in the logs of the Hospital Management System. The main goals include detecting inconsistencies in data, security breaches and any abnormalities that could potentially compromise the confidentiality, availability and integrity of healthcare information. The paper is organized as follows: Section 2 addresses the problem statement, discussing the challenges faced by HMS. Section 3 provides a literature review, summarizing recent methodologies in HMS. Section 4 details the methodology, including data collection, preprocessing, feature engineering and the application of machine learning techniques for anomaly detection. Section 5 presents a classification report, evaluating the performance of machine learning models. Finally, Section 6 concludes the paper by summarizing the findings and discussing the implications for enhancing HMS through log-based testing.

II. PROBLEM STATEMENT

"In the healthcare industry, it is essential to have a transparent management system in hospitals to ensure patient care and resource utilization. However, hospitals often face challenges such as data inconsistencies, vulnerabilities in security measures and operational inefficiencies, within their management systems. These issues can result in errors, delays and potential security breaches. Therefore, this project aims to tackle these problems by utilizing machine learning techniques to analyze and optimize the log data of hospital management systems. This will help enhance data integrity, improve security measures and increase efficiency while maintaining a standard of patient care."

III. LITERATURE REVIEW

Hospital Management Systems (HMS) have become essential for improving patient care, operational efficiency, and compliance with regulations. The increasing complexity of HMS data has made manual testing methods impractical, necessitating advanced techniques for analyzing and predicting system performance. Machine learning (ML) offers powerful tools to analyze system records, enabling anomaly detection, system improvement, and problem prediction. This review examines ML methods applied to log-based testing in HMS, focusing on techniques like anomaly detection and predictive modeling.

One approach, decision trees, has shown effectiveness in failure diagnosis by analyzing runtime data. This method successfully identified 13 out of 14 true failure causes with minimal false positives and demonstrated practical benefits when applied to eBay's production system. Additionally, a cost-benefit analysis emphasized the advantages of automated diagnosis over manual methods. While other classifiers may offer better accuracy, decision trees are favored for their interpretability [1]. The anomaly detection framework consists of four main steps. First, system logs, which contain vital information like status and runtime events, are collected. Second, log parsing converts raw messages into structured data by dividing them into constant and variable segments. Third, feature extraction breaks logs into sequences, generating feature vectors that form a feature matrix. Lastly, this matrix is fed into machine learning models trained to detect anomalies by identifying abnormal patterns in incoming log sequences, automating anomaly detection in large systems [2]. Hospital Management Systems

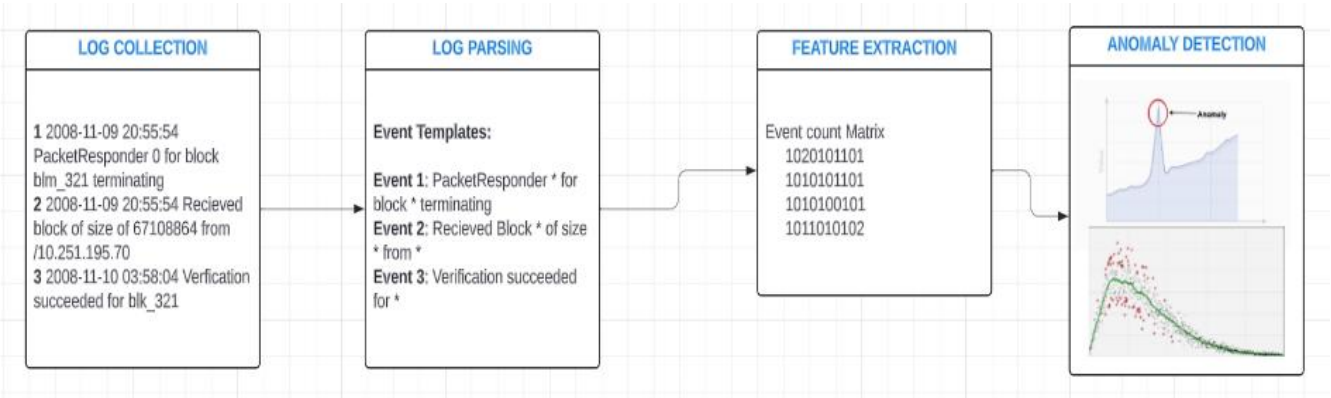

Fig 1. Framework to process Patient Logs at Healthcare Systems

increasingly rely on machine learning to analyze system logs, as manual anomaly detection is not feasible. Automated methods, such as the decision tree approach, have proven effective in diagnosing failures and reducing false positives. This approach, tested on eBay's systems, also demonstrates the cost efficiency of automated diagnostics. The anomaly detection framework automates log analysis, improving reliability and performance in healthcare infrastructure [9].

Anomaly detection in system logs has seen significant advancements with the rise of machine learning and the increasing complexity of systems. A detailed study [4] highlights the use of various machine learning algorithms to identify anomalies, focusing on preprocessing and feature extraction techniques to improve detection accuracy. Murphy and Larsson [5] explore generating embeddings for log messages to capture contextual information, enhancing anomaly detection. Good et al. (2023) further investigate feature learning for developing interpretable and efficient decision trees for this purpose.

This research is significant for its advancements in decision tree algorithms. Huo et al. (2023) underscore the importance of log sequence synthesis in evaluating and enhancing detection algorithms, which aligns well with our focus on real-world log data. Collectively, these studies provide valuable insights into anomaly detection in system logs, utilizing various machine learning techniques and frameworks to improve detection capabilities.

IV. METHODOLOGY

We will collect log files from hospitals as our primary data source, followed by rigorous data preprocessing to ensure data quality and consistency. We will employ machine learning techniques for anomaly detection, focusing on developing and training models capable of identifying abnormal behaviors and patterns. To evaluate the performance of these models, we will establish an experimental setup that utilizes appropriate metrics and benchmarks. Anomaly detection is crucial for incident management in large-scale systems, as it aims to uncover abnormal behaviors in a timely manner, leveraging logs that record detailed runtime information. The log analysis process involves four main steps: log collection, log parsing, feature extraction, and anomaly detection. Anomaly detection methods can be broadly classified into supervised and unsupervised techniques, depending on the data involved and the machine learning methods used. Hospital system logs, which contain timestamps and event messages, will be parsed into structured event templates with specific parameters. Following this, we will encode the parsed logs into numerical feature vectors suitable for machine learning applications, utilizing various grouping techniques such as fixed windows, sliding windows, and session windows. Finally, the resulting feature matrix will be fed into machine learning models for training, facilitating the generation of an effective anomaly detection model [2].

A. *Framework for Anomaly Detection System*

1) Data Collection and Preprocessing: Our journey starts with data collection from Shamsi Hospital, focusing on acquiring comprehensive log files, including billing records, admission details, and payment transactions. This raw data serves as the primary input for our anomaly detection system. We then embark on rigorous data preprocessing to ensure data quality and consistency. The preprocessing stage involves:

2) Data Cleaning: Removing outliers and handling missing values.

*3) Data Filtering:*Removing irrelevant information to ensure data quality and consistency.

4) Feature Engineering: Once the data is cleaned and prepared, we proceed with feature engineering. The initial dataset had 60 columns representing various aspects of hospital operations. Through this process, we identified the most relevant features for anomaly detection, reducing the columns to 7. This involved selecting features that significantly enhance the model's performance while eliminating redundant or irrelevant data. The final set includes one class column, "Anomaly," indicating whether a record is anomalous.

Table I: Relevant Features from Dataset

ID	*PT_Admission*	*PT_Discharge*		*PT_ChargeAmount*	*PT_BillDate*	*Anomaly*
1	*16*	*19*		*21*	*22*	*27*

B. *Decision Tree Approach*

The anomaly detection system is based on a decision tree algorithm, chosen for its interpretability and effectiveness in handling structured data.

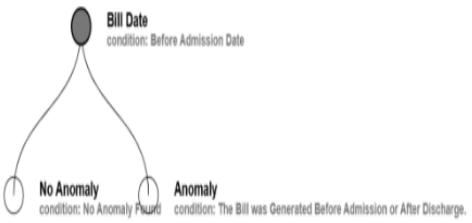

Fig. 2. An example of Decision Tree for Anomaly Detection

Here are the key steps involved in our implementation:

1) Model Training: We split the preprocessed dataset into training and testing sets. The training set is used to build the decision tree model. The algorithm recursively splits the data based on feature values that result in the most significant information gain, creating a tree structure where each node represents a decision point.

2) Feature Selection: During the training process, the decision tree algorithm automatically selects the most relevant features, optimizing the model's ability to identify anomalies. This step enhances the accuracy and efficiency of the anomaly detection process.

3) Model Evaluation: After training, the model is validated using the testing set. We evaluate its performance using key metrics such as accuracy, precision, recall, and F1-score. The F-score is defined as the harmonic mean of precision and recall.

$$\text{RECALL} = \frac{TP}{TP+FN} \quad (1)$$

$$\text{PRECISION} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{F1-SCORE} = 2 \times \frac{precision \times \text{recall}}{precision+recall} \quad (3)$$

Table II. Illustrating all equations used in the ML algorithms

	precision	recall	f1-score	Support
0	0.86	0.88	0.87	1634
1	0.82	0.80	0.81	1180
accuracy			0.85	2814
macro avg	0.84	0.84	0.84	2814
weighted avg	0.84	0.85	0.84	2814

Detecte

V. CLASSIFICATION REPORT

To evaluate the effectiveness of our models, we computed several performance metrics including accuracy and the classification report. Additionally, we calculated the Gini index as a effectiveness in binary classification. The Gini index is a widely used metric in machine learning that quantifies the degree of class imbalance in the dataset. For our dataset, the Gini Index is as follows:

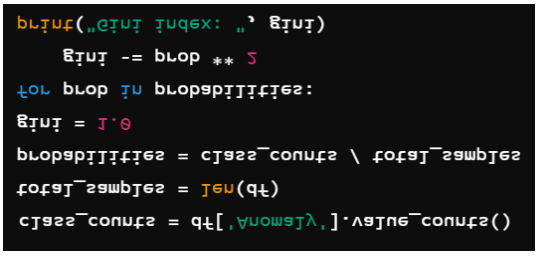

Fig. 3. Code Snippet of Gini Index Calculation

The Gini index for our dataset was 0.4451, indicating a moderate level of class imbalance (Gini index: **0.4451**).

A. *Visualization and Reporting*

Detected anomalies are displayed in a user-friendly, dynamic table on the frontend, allowing administrators to easily interpret the results. Additionally, an interactive dashboard provides comprehensive insights into the frequency of anomalies, patient statistics, and system performance metrics. These visual tools enhance the administrators' ability to monitor and manage hospital operations effectively.

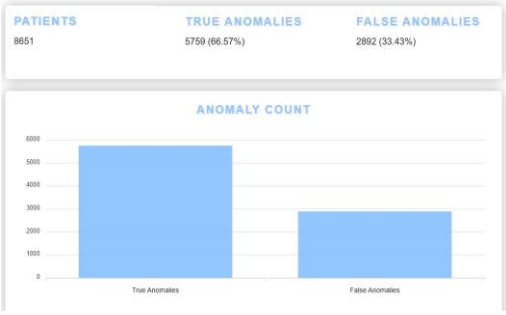

Fig. 4. Anomaly Detection Dashboard

VI. CONCLUSION

The integration of machine learning for log-based testing in healthcare management systems presents a promising avenue to enhance data integrity, security, and operational efficiency. By leveraging techniques like Decision Trees, we can proactively identify anomalies, ensuring the seamless execution of hospital functions and elevating the standard of patient care. This innovative approach signifies a transformative step towards optimizing hospital management in the

era of healthcare digitization, while also recognizing the undeniable efficiency of automated testing over manual methods.

A. Results and Implications

Our research highlights the importance of log-based testing in identifying security vulnerabilities, data inconsistencies, and operational errors. Our system's ability to analyze log entries using machine learning techniques ensures that the hospital system executes expected functions, identifies potential issues, and verifies user. compliance with prescribed workflows. In order to select the most efficient model for predictions, comparative analysis was done of various models on the basis of their accuracy scores.

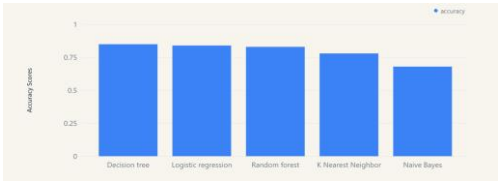

Fig. 5. Bar chart of corresponding accuracies for algorithms tested for Hospital Management System

On the basis of which decision tree was selected. After further enhancements in the model, for example hyper parameter tuning and handling imbalanced data, accuracy score rose up to 0.87. Precision of anomalies and non-anomalies came out as 0.92 and 0.77 respectively. The F1-score (0.90) is high, suggesting that the model performs very well in identifying anomalies. The Gini Index is 0.4451 that shows moderate impurity but it does not have any negative effect on the model's metrics. In conclusion, the use of machine learning, particularly Decision Trees, for log-based testing in healthcare systems shows strong potential to improve system reliability and patient care through effective anomaly detection and operational efficiency.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Dr. Abdul Aziz, Department Head of Software Engineering at FAST NUCES Karachi, for his invaluable guidance and support throughout this research. We extend our special thanks to Shamsi Hospital for providing access to the dataset, which was essential for conducting this study. We also appreciate the encouragement and insightful feedback from our colleagues in the Software Engineering department at FAST NUCES Karachi, as well as the resources and facilities provided by the institution, which greatly facilitated the successful completion of this project. Finally, we thank the organizing committee of the INIMC 2024 IEEE Conference at Saleem Habib University for the opportunity to present this work and for their support in bringing together researchers in this field.

VII. REFRENCES

[1] Chen, M., Zheng, A. X., Lloyd, J., Jordan, M. I., & Brewer, E. (2021). *Failure diagnosis using decision trees*. In International Conference on Autonomic Computing, 2021. Proceedings (pp. 36-43). IEEE.

[2] He, S., Zhu, J., He, P., & Lyu, M. R. (2020). Experience Report: System Log Analysis for Anomaly Detection. 2020.

IEEE 27th International Symposium on Software Reliability Engineering (ISSRE). doi:10.1109/issre.2016.21.

[3] Zhu, J., He, S., Liu, J., He, P., Xie, Q., Zheng, Z., & Lyu, M. R. (2019). Tools and Benchmarks for Automated Log Parsing. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP).
[4] Moresová, Eva. "Anomaly Detection in System Log Files Using Machine Learning." Master's Thesis, Brno University of Technology, Faculty of Information Technology, 2024.

[5] A. Murphy and D. Larsson, "Towards Automated Log Message Embeddings for Anomaly Detection," Master's Thesis, Dept. of Automatic Control, Lund University, Lund, Sweden, 2024.

[6] J. H. Good, T. Kovach, K. Miller, and A. Dubrawski, "Feature Learning for Interpretable, Performant Decision Trees," in *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, USA, 2023.

[7] T. Anttila, "Developing a log file analysis tool: A machine learning approach for anomaly detection," M.S. thesis, Faculty of Information Technology and Electrical Engineering, University of Oulu, Oulu, Finland, 2020.

[8] Y. Huo, Y. Li, Y. Su, P. He, Z. Xie, and M. R. Lyu, "AutoLog: A Log Sequence Synthesis Framework for Anomaly Detection,",, Aug. 2023.

[9] L. Breiman, J. H.Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Wadsworth, 1984.

[10] A. Brown, G. Kar, and A. Keller. An Active Approach to Characterizing Dynamic Dependencies for Problem Determination in a Distributed Environment. In Seventh IFIP/IEEE International Symposium on Integrated Network Management, Seattle, WA, May 2001.

[11] M. Chen, A. Accardi, E. Kıcıman, J. Lloyd, D. Patterson, A. Fox, and E. Brewer. Path-based Failure

and Evolution Management. In Proceedings of the First Symposium on Networked Systems Design and Implementation (NSDI), San Francisco, CA, 2004.

[12] M. Chen, E. Kıcıman, E. Fratkin, E. Brewer, and A. Fox. Pinpoint: Problem Determination in Large, Dynamic Internet Services. In International Computer Performance and Dependability Symposium, 2002.

[13] J. Choi, M. Choi, and S. Lee. An alarm correlation and fault identification scheme based on OSI managed object classes. In IEEE International Conference on Communications, Vancouver, BC, Canada, 1999.

[14] G. F. Cooper. A simple algorithm for efficiently mining observational databases for causal relationships. Journal of Data Mining and Knowledge Discovery, 1(1-2):245–271, 1997.

[15] H. P. Corporation. HP Openview. http://www.hp.com/ openview/index.html.

[16] G. H. John, R. Kohavi and K. Pfleger. Irrelevant features and the subset selection problem. Machine Learning: Proceedings of the Eleventh International Conference, pages 121– 129, 1994.

[17] G. H. John, R. Kohavi and K. Pfleger. Irrelevant features and the subset selection problem. Machine Learning: Proceedings of the Eleventh International Conference, pages 121– 129, 1994.

[18] B. Gruschke. A new approach for event correlation based on dependency graphs. In 5th Workshop of the OpenView University Association, 1998.

[19] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. JMLR Special Issue on Variable and Feature Selection, 3(Mar):1157-1182, 2003.

[20] I. H. Witten and E. Frank. Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann.

[21] I. Rish and M. Brodie and N. Odintsova and S. Ma, G. Grabarnik. Real-time problem determination in distributed systems using active probing. In Network Operations and Management Systems, 2004.

[22] I. Rish, M. Brodie, and S. Ma. Accuracy vs. efficiency tradeoffs in probabilistic diagnosis. In AAAI-2002, Edmonton, Alberta, Canada, 2002.

[23] IBM. Tivoli Business Systems Manager, 2001. http:// www.tivoli.com. [17] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.

[24] J. Pearl. Causality: Models, Reasoning, and Inference. Cambridge University Press, 2000.

[25] M. Steinder and A. Sethi. End-to-end service failure diagnosis using belief networks. In Network Operations and Management Symposium, 2002.

[26] A. Yemini and S. Kliger. High speed and robust event correlation. IEEE Communication Magazine, 34(5):82–90, May 1996.

[27] R. Venkatakrishnan and M. A. Vouk. Diversity-based detection of security anomalies. In HotSoS'14: Proc. of the 2014 Symposium and Bootcamp on the Science of Security, page 29. ACM, 2014.

[28] W. Xu, L. Huang, A. Fox, D. Patterson, and M.I. Jordon. Detecting large-scale system problems by mining console logs. In SOSP'09: Proc. of the ACM Symposium on Operating Systems Principles, 2009.

[29] D. Yuan, S. Park, P. Huang, Y. Liu, M. Lee, X. Tang, Y. Zhou, and S Savage. Be conservative: enhancing failure diagnosis with proactive logging. In OSDI'12: Proc. of the 10th USENIX Conference on