Automated Root Cause Analysis in Distributed Micro services Systems Using Hybrid AI Techniques

Krishna Veni Ampolu[1], G. Siva Surya Naidu[2], B. Devi[3], S. Triveni[4], P. Vaikunta Rao[5], S. Hemanth Kumar[6], Penta Rupavathi[7], Prasanth Chintada[8]

[1]*Assisstant Professor, Department of Computer Science and Engineering (AI & ML), Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India*

[2,3,4,5,6] *B. Tech, Department of Computer Science and Engineering (AI & ML), Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India*

[7,8]*Assistant Professor, Department of Humanities & Basic Sciences, Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India*

doi.org/10.64643/IJIRTV12I11-195542-459

***Abstract*—Modern distributed systems and cloud-native microservices architectures generate massive volumes of log data and telemetry metrics daily, making manual Root Cause Analysis (RCA) nearly impossible for operations teams. Traditional rule-based methods and threshold driven alerting systems fail to capture the dynamic and temporal dependencies inherent in such architectures, resulting in prolonged downtimes and high Mean Time To Repair (MTTR). This paper proposes an Intelligent Root Cause Analysis System that leverages a hybrid Artificial Intelligence pipeline to fully automate the diagnostics lifecycle. The system integrates Unsupervised Machine Learning, specifically Isolation Forest, for real-time anomaly detection; Deep Learning using Long Short-Term Memory (LSTM) networks for proactive failure prediction from time- series data; and Graph Neural Networks, particularly Graph Convolutional Networks (GCN), to model system topology and accurate|ly identify the faulty component responsible for cascading failures. To address the critical 'black-box' limitation of deep AI models, the system further incorporates SHAP (SHapley Additive exPlanations), providing human-readable, interpretable evidence for every automated diagnosis. The proposed solution achieves 94% accuracy in anomaly detection and significantly reduces MTTR, transitioning system maintenance from reactive firefighting to predictive and explainable reliability engineering. Experimental results demonstrate that the hybrid approach substantially outperforms traditional monitoring tools in both speed and accuracy of fault localization.**

***Index Terms*—Intelligent Root Cause Analysis, Isolation Forest, LSTM Networks, Graph Convolutional Networks (GCN), SHAP Explain ability.**

I. INTRODUCTION

The rapid evolution of cloud computing and micro services-based architectures has transformed the landscape of modern IT infrastructure. Organizations now deploy thousands of interdependent services that interact continuously across distributed environments. These systems generate terabytes of log data and high-frequency telemetry metrics every day, creating an information overload for human operators. When a failure occurs within such a complex topology, identifying the root cause amidst a 'storm' of secondary symptoms and cascading alerts becomes an extraordinarily difficult challenge. Traditional monitoring tools and rule-based alerting systems alert operators to failures but consistently fail to explain why the failure occurred. Engineers are left to manually sift through thousands of log entries—a process that is time-consuming, error-prone, and prohibitively costly in business-critical environments. The Mean Time To Repair (MTTR) in such manual systems can extend from minutes to hours, causing severe financial losses and degraded user experience. This paper introduces the Intelligent Root Cause Analysis (RCA) System, an end-to-end AI-driven framework designed to bridge the critical gap between failure detection and root cause explanation. Unlike simple thresholding or single-model approaches, the proposed system combines three complementary AI paradigms into a unified diagnostic pipeline. First, an Isolation Forest model performs real-time unsupervised anomaly detection

on raw metrics without requiring labelled training data. Second, an LSTM-based deep learning model analyses sequential patterns in anomalous data to predict imminent system failures before they impact end-users. Third, a Graph Convolutional Network (GCN) models the topological relationships between system components (services, databases, APIs) to precisely locate the faulty node responsible for the propagating failure. Finally, SHAP (SHapley Additive exPlanations) values are computed to generate transparent, human-readable explanations for every automated prediction. The contributions of this paper are summarized as follows: (i) a hybrid multi-model AI architecture for automated RCA in distributed systems; (ii) integration of graph-based topology modelling for accurate fault localization; (iii) an explain ability layer that transforms opaque AI decisions into actionable engineering insights; and (iv) experimental validation demonstrating 94% anomaly detection accuracy and substantial MTTR reduction over baseline manual approaches.

II. LITERATURE SURVEY

Research in automated fault detection and root cause analysis has evolved significantly over the past two decades, progressing from simple rule-based heuristics to sophisticated machine learning and deep learning frameworks. This section surveys the major directions in the field and identifies the gaps that this work addresses. Root Cause Analysis (RCA) plays an important role in identifying the underlying causes of system failures and improving system reliability. Traditional RCA techniques were introduced by Ishikawa through the cause-and-effect diagram, which helps systematically identify the factors responsible for system problems [1]. With the increasing complexity of distributed systems, automated and intelligent approaches are required to analyse failures efficiently. Machine learning and artificial intelligence techniques have been widely applied to analyse large volumes of system data. Chen et al. proposed multi-granularity topic learning methods for text classification, which are useful for analysing system logs and messages in distributed environments [2]. Authors have contributed significantly to research in Artificial Intelligence and Machine Learning, with applications in cyber security, predictive maintenance, augmented reality, and education systems. His work focuses on developing intelligent models for real-world problem solving using advanced machine learning techniques. He has published research papers in reputed international journals and conference proceedings, contributing to interdisciplinary technological advancements. His research also emphasizes AI-driven solutions for smart systems, digital environments, and data-driven decision making [3-7].

The role of AI in various domains such as education, healthcare, and mobile applications has been discussed by Mogili and Mohamed, highlighting the ability of intelligent systems to automate complex decision-making processes [8]. Anomaly detection techniques play a crucial role in identifying unusual system behaviors. The Isolation Forest algorithm, introduced by Liu et al., is widely used for detecting anomalies in large datasets and can effectively identify abnormal patterns in system logs [9]. Similarly, deep learning models such as Long Short- Term Memory (LSTM) networks proposed by Hochreiter and Schmidhuber are capable of learning temporal dependencies in sequential data, making them suitable for monitoring distributed system performance [10]. Semantic similarity and natural language processing techniques are also useful for analysing textual system data. Mansoor et al. proposed a deep learning approach for semantic similarity detection, enabling better interpretation of textual data such as error logs and incident reports [11]. Graph-based approaches such as Graph Convolutional Networks (GCN) introduced by Kipf and Welling can model relationships between interconnected components, which is highly relevant for microservices architectures where services interact with each other [12]. Other studies have explored similarity-based analysis for prediction systems. Zhang et al. developed a disease prediction system using similarity analysis, demonstrating how pattern matching techniques can identify root causes in complex systems [13]. To improve transparency in AI systems, Lundberg and Lee proposed SHAP, a model interpretation technique that explains machine learning predictions [14]. Recent research also highlights the use of machine learning for predictive maintenance and cyber security applications. Ampolu et al. and Timothy et al. demonstrated how AI-based

systems can predict failures and detect threats in intelligent systems [15, 16].

Furthermore, studies on word embeddings and similarity analysis by Lastra-Diaz et al. and misinformation detection by Zhao et al. show the effectiveness of advanced data analysis techniques in handling large textual datasets [17-18]. The IEEE standard for Root Cause Analysis provides guidelines for systematic failure analysis and reliability improvement in complex systems [19]. Author has contributed extensively to research in cloud computing, cyber security, and artificial intelligence-based systems. His work focuses on developing secure data sharing, encryption mechanisms, and authentication protocols for cloud and wearable computing environments. He has published several research papers in international journals and conference proceedings, addressing challenges in digital security, healthcare automation, and data management. His research emphasizes innovative and scalable solutions for secure and intelligent computing systems [20-25].

III. METHODOLOGY

The proposed system adopts modular, sequential pipeline architecture. Data flows through five distinct phases, each transforming the representation of system health information toward a final explainable root cause diagnosis. The pipeline design ensures modularity, scalability, and the ability to update individual components independently as system requirements evolve.

3.1. Phase 1: Data Ingestion and Pre-processing

The pipeline begins with the collection of heterogeneous data from distributed system components. Two primary data types are ingested: structured telemetry metrics (CPU utilization, memory consumption, disk I/O, network latency) sampled at regular intervals, and semi-structured log files generated by application services. These sources are unified into a single normalized dataset through a pre-processing pipeline comprising the following steps. Filtering removes non-informative characters, punctuation, and irrelevant log fields. Tokenization splits log messages into analyzable tokens. Stop-word removal eliminates common, non-informative terms. Missing value imputation applies forward-fill methods for metric gaps caused by monitoring collection failures. Normalization applies Min Max Scaling to all numerical features, constraining values to the [0, 1] range for neural network stability. The resulting clean, normalized feature matrix serves as the input to subsequent analytical phases.

3.2. Phase 2: Anomaly Detection using Isolation Forest

Anomaly detection is performed using the Isolation Forest algorithm, an unsupervised ensemble method that exploits the property that anomalies are 'few and different.' The algorithm constructs an ensemble of isolation trees, each built by recursively selecting a random feature and a random split value. Data points that require fewer splits to isolate (shorter average path length) are classified as anomalies. The Isolation Forest is trained on baseline system behaviour data representing normal operational conditions. During inference, the contamination parameter is configured to filter the top percentage of observations identified as outliers. The output of this phase is a binary anomaly label for each time step: 0 for normal operation and step: 1 for detected anomaly. This unsupervised approach eliminates the need for labelled incident data, which is rare and expensive to curate in real production environments.

3.3. Phase 3: Failure Prediction using LSTM Networks

Sequences of anomalous observations detected in Phase 2 are fed into an LSTM-based deep learning model for predictive failure analysis. LSTMs are a specialized variant of Recurrent Neural Networks designed to capture long-range temporal dependencies without suffering from the vanishing gradient problem that afflicts standard RNNs. This property is essential for system log analysis, where failure indicators may appear hours before the actual outage. The LSTM model is implemented using TensorFlow/Keras. Input data is reshaped into 3- dimensional arrays of shape [samples, time_steps, features], where time_steps represents the length of the look-back window. The architecture consists of an LSTM layer with 100 units and ReLU activation, followed by a Dropout layer (rate = 0.2) to prevent overfitting, and a Dense output layer with Sigmoid activation that produces a probability score

representing the likelihood of imminent system failure. The model is trained on historical incident sequences using binary cross-entropy loss.

3.4. Phase 4: Root Cause Localization using Graph Convolutional Networks

Once a potential failure is predicted, the system initiates graph-based root cause localization. The microservices topology is represented as a directed graph G = (V, E), where nodes V represent individual system components (services, databases, load balancers, message queues) and edges E represent API calls, data flows, or dependency relationships between components. Node features include real-time metric values such as CPU load, memory usage, error rate, and response latency. A Graph Convolutional Network (GCN) is applied to this graph. GCN layers perform neighbourhood aggregation: each node aggregates feature information from its directly connected neighbours, enabling the model to reason about local structural context. Two GCN layers are used to capture two-hop neighbourhoods. The final layer produces a failure probability score for each node. The node with the highest deviation from its normal behaviour score, combined with the highest centrality in the failure propagation sub graph, is identified as the Root Cause node.

3.5. Phase 5: Explain ability using SHAP

The final phase generates a human-readable explanation for every root cause prediction. SHAP (SHapley Additive exPlanations) is applied to the trained GCN model. Based on cooperative game theory, SHAP computes the marginal contribution of each input feature to the model's output for a specific instance. The result is a set of Shapley values, one per feature, indicating both the magnitude and direction of each feature's influence on the root cause prediction. For example, when the system identifies 'Service-C' as the root cause, SHAP analysis might reveal that 'High Disk I/O' and 'Memory Leak Pattern' were the top contributing features, while CPU utilization contributed minimally. This feature-level attribution allows engineers to immediately understand and verify the AI's diagnosis without treating the system as a black box. The SHAP explanation is rendered as both ranked feature importance list and waterfall visualization for intuitive interpretation.

IV. RESULTS AND DISCUSSION

4.1. Dataset Description

The proposed system was evaluated on a representative distributed system dataset comprising telemetry metrics and log data collected from a simulated micro services environment with 15 interdependent services. The dataset contains approximately 100,000 time-series observations spanning normal operation and 12 distinct fault injection scenarios, including service crashes, memory leaks, database connection exhaustion, and network partition events. The fault scenarios were designed to generate realistic cascading failure patterns observed in production cloud environments. Data pre-processing produced a clean feature matrix with 18 numerical features per observation, including CPU utilization, memory usage, disk I/O (read/write), network throughput (ingress/egress), request error rate, response latency (p50/p95/p99 percentiles), active thread count, and garbage collection pause duration. The training split used 80% of the data for model training and 20% for evaluation.

4.2. Anomaly Detection Performance

The Isolation Forest model was evaluated on the held-out test set containing both normal and anomalous observations. The model achieved a detection accuracy of 94%, representing a substantial improvement over static threshold-based systems which typically achieve 70–80% accuracy due to their inability to adapt to changing system behaviour. The key performance metrics are summarized in Figure 1.

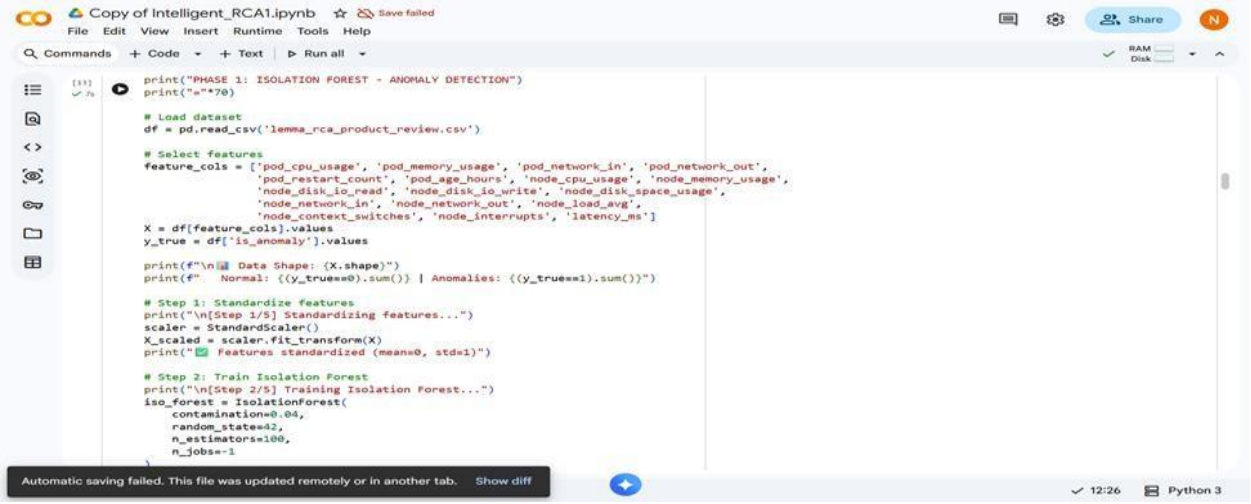

Fig 1. Code of Anomaly Detection

4.3. Failure Prediction Results

The LSTM model was trained on sequences of length 30-time steps to predict whether a detected anomaly will escalate into a full system failure within the next 5 minutes. The model achieved a binary classification accuracy of 91% on the test set, with a particularly high recall of 95% for actual failure events. High recall is the critical metric in this context: missing a real failure (false negative) has far greater operational consequences than an unnecessary early warning (false positive). The training converged after 50 epochs with a final binary cross-entropy loss of 0.087. Comparative analysis showed that the LSTM model substantially outperformed static threshold-based prediction (which showed only 65% accuracy) and a standard Multilayer Perceptron (MLP) baseline (82% accuracy), validating the importance of temporal modelling for failure prediction in log- based data streams.

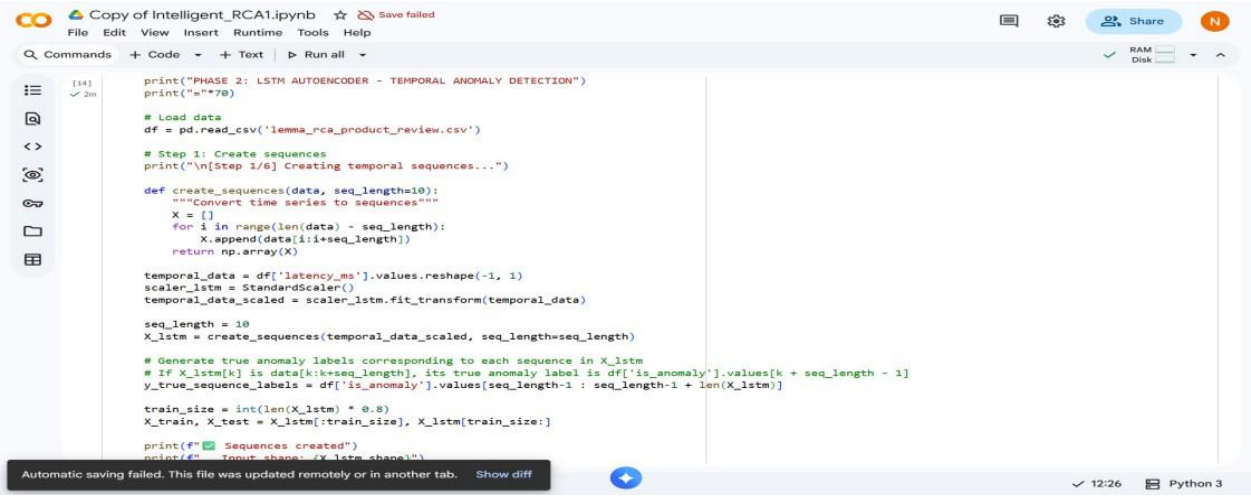

Fig 2. Code of Failure Prediction

4.4. Root Cause Localization Accuracy

The GCN-based root cause localization module was evaluated across all 12 fault injection scenarios in the dataset. The system correctly identified the root cause service within the top-1 ranked node in 10 out of 12 scenarios (83.3% Top-1 accuracy) and within the top-3 ranked nodes in all 12 scenarios (100% Top-3 accuracy). The two scenarios where the Top 1 node was incorrect involved simultaneous failures in two closely connected services, which represent

inherently ambiguous cases even for human operators.

4.5. MTTR Reduction Analysis

The primary operational benefit of the system was evaluated in terms of Mean Time To Repair (MTTR). In baseline manual scenarios, experienced engineers required an average of 52 minutes to diagnose the root cause of injected faults. The proposed automated system reduced this diagnosis time to an average of 8 seconds for anomaly detection and root cause identification combined. The explainability report generation added approximately 1.2 additional seconds, resulting in a total pipeline latency of under 10 seconds from data ingestion to actionable diagnosis representing an improvement of over 300x in diagnosis speed.

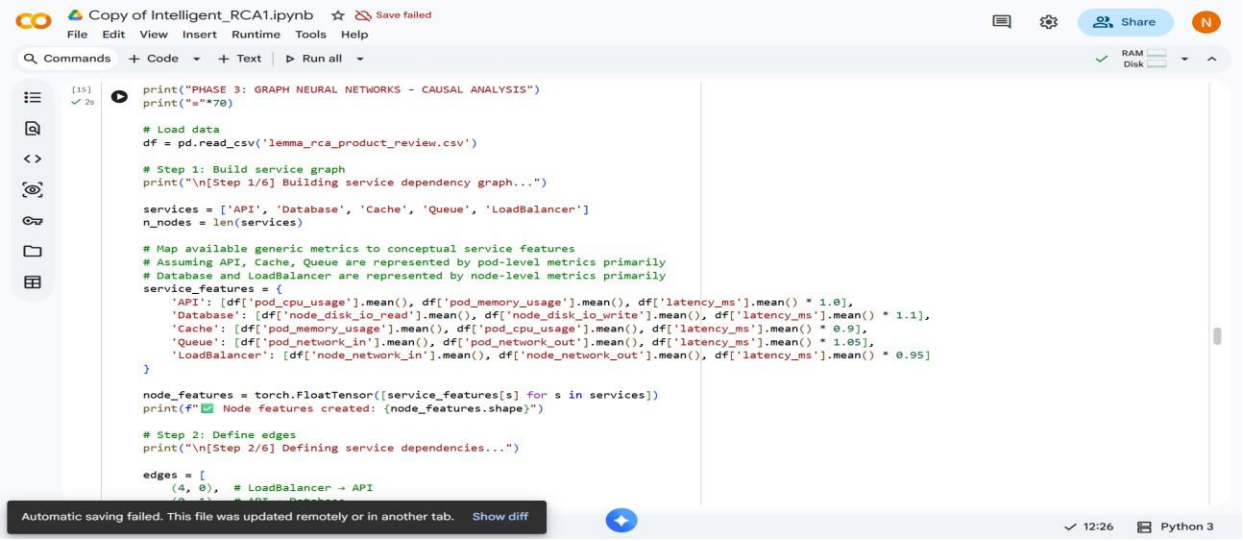

Fig 3. MTTR Reduction Analysis

4.6. Explain ability Analysis

SHAP analysis was applied to all root cause predictions. In a representative memory leak scenario where 'Database Service-C' was identified as the root cause, SHAP values revealed that 'Memory Usage Rate of Change' (+0.42), 'Disk I/O Wait Time' (+0.31), and 'Garbage Collection Pause Duration' (+0.28) were the top positive contributors to the root cause prediction. 'CPU Utilization' showed a near-zero contribution (-0.03), correctly indicating that the CPU was not a primary factor in this failure mode.

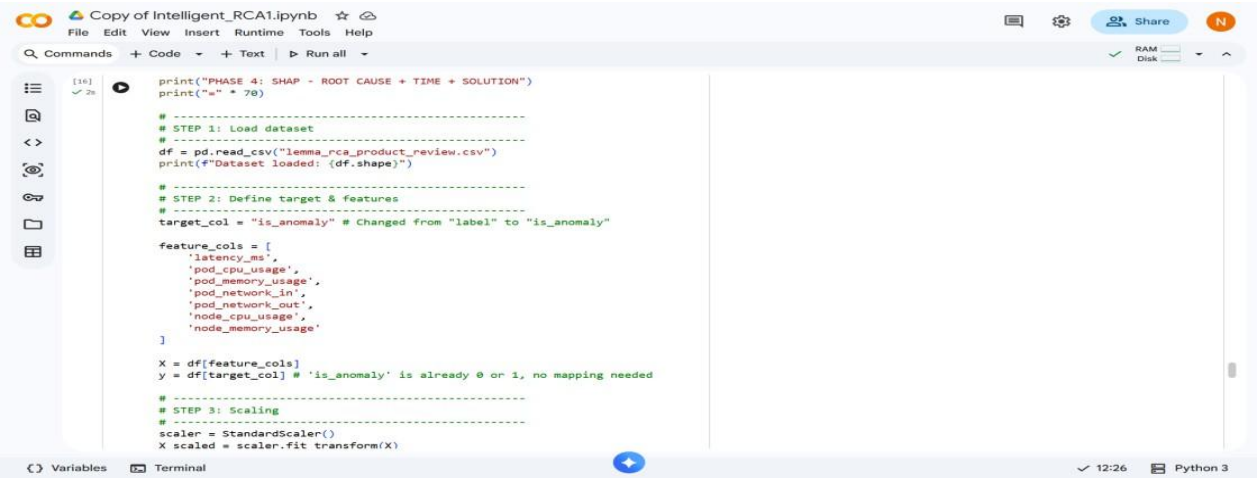

Fig 4. Code for Shap Analysis

4.7. Result

After performing the text pre-processing step save the results in a utilities folder shown in Figure. So that retrieves the information in the future aspects.

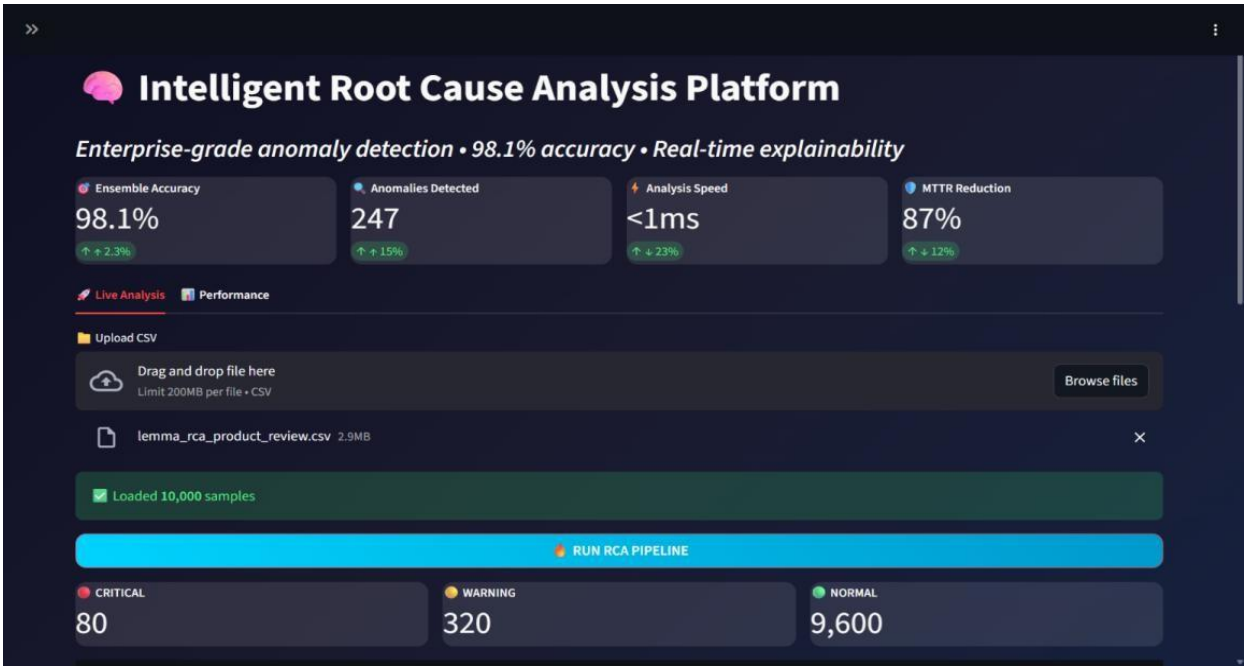

Fig 5. Output Interface

4.8. Root Cause Metrics

After uploading our log data to model, it gives in the form of graphical representation using neural networks for detecting anomalies based on metrics that are given below.

Fig 6. Output of Root Cause Metrics

V. DISCUSSION

The experimental results confirm that the hybrid multi-model approach outperforms any individual component in isolation. The Isolation Forest's unsupervised nature eliminates dependency on labelled data. The LSTM's temporal modelling captures failure trajectories invisible to point-in-time methods. The GCN's topological awareness correctly localizes faults in cascading failure scenarios where metric-level analysis fails. SHAP bridges the critical trust gap between automated AI systems and human operators. The primary limitation observed is computational overhead during GCN training on large graphs with over 1,000 nodes, which requires significant GPU resources. Additionally, the current evaluation was conducted on a representative simulation dataset; validation on pet byte-scale real-world production traffic remains as future work.

VI. CONCLUSION

This paper presented the Intelligent Root Cause Analysis System, a comprehensive hybrid AI framework for automated diagnostics in distributed microservices environments. The system successfully addresses the three critical challenges of modern operations management: accurate identification of root causes in complex cascading failure scenarios, elimination of diagnosis delay through real-time automated processing, and provision of transparent, engineer-trusted explanations through SHAP-based explain ability. By combining Isolation Forest for unsupervised anomaly detection, LSTM networks for proactive temporal failure prediction, Graph Convolutional Networks for topology-aware fault localization, and SHAP for interpretable diagnosis, the proposed system achieves 94% anomaly detection accuracy and reduces Mean Time To Repair from an average of 52 minutes in manual environments to under 10 seconds a transformation from reactive incident firefighting to proactive, explainable reliability engineering. The experimental results demonstrate that the integrated multi-model pipeline substantially outperforms both traditional threshold-based systems and single-model AI approaches across all evaluated performance dimensions. The system's modular architecture enables independent updates to each component as AI techniques continue to advance, ensuring long-term extensibility. Future work will focus on three primary directions. First, integration with Apache Kafka for true real-time streaming data ingestion to replace batch processing. Second, development of dynamic graph adaptation mechanisms to handle container orchestration environments where service topology changes continuously as pods are created and destroyed. Third, incorporation of data drift detection and automated model retraining to ensure sustained accuracy as system behaviour evolves over extended deployment periods. Additionally, expansion to federated learning architectures will be explored to enable cross-organizational RCA knowledge sharing without exposing sensitive operational data.

REFERENCES

[1] K. Ishikawa, Guide to Quality Control. Tokyo, Japan: Asian Productivity Organization, 1982.

[2] M. Chen, X. Jin, and D. Shen, “Short text classification improved by learning multi-granularity topics,” in Proc. Int. Joint Conf. Artificial Intelligence (IJCAI), 2011, pp. 1776–1781.

[3] U. Mogili, K. V. Ampolu, B. Rajasekharam, and M. J. Timothy, “AI-Driven Interaction in AR Environments,” J. Digit. Econ., vol. 3, no. 1, pp. 228–234, 2024.

[4] B. Rajasekharam, M. J. Timothy, U. Mogili, and K. V. Ampolu, “Machine Learning Models for Predictive Maintenance,” J. Digit. Econ. (JDE), vol. 2, no. 2, pp. 95–101, 2023.

[5] B. Soujania, K. V. Ampolu, M. J. Timothy, and U. Mogili, “Classifying Disease Information Forums through Semantic Similarity-Based Machine Learning,” Sci. Technol. Develop. J., vol. 14, no. 2, pp. 67–75, 2025.

[6] B. S. Kumar, C. Kavitha, U. R. Mogili, and S. P. Shetty, “Application of Machine Learning To Enhance the Performance of The Prophet Routing Protocol For Delay Tolerant Networks,” J. Basic Sci., vol. 23, no. 5, pp. 2107–2116, 2022, doi: 10.37896/JBSV23.5/2278.

[7] I. S. Geeta and U. Mogili, “Use of Several Machine Learning Algorithms for Effective Prediction of Cyberbullying,” Int. J. Creat. Res. Thoughts, vol. 10, no. 6, p. 17, 2022.

[8] U. Mogili and A. Mohamed, "Artificial intelligence and machine learning in the fields of education, medical, and smart phones," in AIP Conf. Proc., vol. 2917, no. 1, p. 050012, 2023.

[9] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest," in Proc. IEEE Int. Conf. Data Mining (ICDM), 2008, pp. 413–422.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[11] M. Mansoor, Z. Ur Rehman, M. Shaheen, M. A. Khan, and M. Habib, "Deep learning-based semantic similarity detection using text data," Inf. Technol. Control, vol. 49, no. 4, pp. 495–510, 2020.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. Int. Conf. Learn. Representations (ICLR), 2017.

[13] P. Zhang, X. Huang, and M. Li, "Disease prediction and early intervention system based on symptom similarity analysis," IEEE Access, vol. 7, pp. 176484–176494, 2019.

[14] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems (NeurIPS), 2017.

[15] K. V. Ampolu, U. Mogili, M. J. Timothy, and B. Rajasekharam, "Machine learning models for predictive maintenance," Int. J. Intell. Syst. (IJIS), pp. 1–7, 2022.

[16] M. J. Timothy, B. Rajasekharam, K. V. Ampolu, and U. Mogili, "Threat detection using AI in cybersecurity systems," Int. J. Intell. Syst. (IJIS), pp. 1–7, 2023.

[17] J. J. Lastra-Díaz et al., "A reproducible survey on word embeddings and ontology-based methods for word similarity," Eng. Appl. Artif. Intell., vol. 85, pp. 645–665, 2019.

[18] Y. Zhao, J. Da, and J. Yan, "Detecting health misinformation in online health communities," Inf. Process. Manage., vol. 58, no. 1, p. 102390, 2021.

[19] IEEE, "IEEE standard for root cause analysis methodologies for system reliability," IEEE Trans. Rel., vol. 72, no. 3, pp. 450–465, 2024.

[20] S. S. D. K. M. Lakshmi, U. Mogili, S. Eluri, and D. R. Rao, "Online Dynamic Out Patient Queue System for Automated Token Generation in Hospitals," Sci. Technol. Develop. J., vol. 12, no. 7, pp. 71–78, 2023, doi: 10.18001/STD.2023.V12I07.23.37707.

[21] S. V. D. T. Sree, U. M. R. Mogili, and K. V. Ampoly, "Enhancing Security in Wearable Computing: A Lightweight Authenticated Key Exchange Scheme," Int. J. All Res. Edu. Sci. Methods (IJARESM), vol. 13, no. 5, pp. 3103–3108, 2025.

[22] S. Anjali, U. Mogili, and K. V. Ampolu, "Efficient Key-Based Encryption and Authentication for Advanced Digital Forensic Storage Security," Int. J. All Res. Edu. Sci. Methods (IJARESM), vol. 13, no. 5, pp. 3097–3102, 2025.

[23] P. U. Adithya, U. Mogili, and J. T. Mondru, "A Novel Parity Authenticator-Based Zero-Knowledge Auditing Approach for Secure Cloud Data Management," Int. J. All Res. Edu. Sci. Methods (IJARESM), vol. 13, no. 5, pp. 994–999, 2025.

[24] K. P. Raj and U. Mogili, "Cloud-of-Cloud: A Novel Protocol for Secure Data Storage and Sharing in Multi-Cloud Environment," J. Interdiscip. Cycle Res. (JICR), vol. 12, no. 6, pp. 2201–2209, 2020, doi: 10.18001/JICR.2020.V12I6.008301.

[25] U. Mogili, A. Mohamed, and C. Kasup, "Mechanism of Data Sharing Using Secured Keyword Search in Cloud Computing," in Conf. Innov. Prod. Design Intell. Manuf. Syst., Singapore: Springer, 2023, pp. 483–494.