SentinelOps: AI-Based System Failure Prediction Using Application Logs

SentinelOps (AI-SFPL)

Dr. M. Jaithoon Bibi#1, Dr. V. Krishna Priya#2, A. Mohammed Arshad#3^3

#1Assistant professor, Department of Computer Science with Cognitive Systems, Sri Ramakrishna College of Arts & Science, Coimbatore, Tamilnadu, India.

#2Assistant professor, Department of Computer Science with Cognitive Systems, Sri Ramakrishna College of Arts & Science, Coimbatore, Tamilnadu, India.

#3Student of Computer Science with Cognitive Systems, Sri Ramakrishna College of Arts & Science, Coimbatore, Tamilnadu, India.

1jaithoonbibi@srcas.ac.in, 2krishnapriya@srcas.ac.in, 3arshadpvt313@gmail.com

Abstract

Modern web applications generate massive volumes of application logs that contain valuable information about system behavior, server performance, and error events. Traditional monitoring tools rely on rule-based alerts or manual log analysis, which detect failures only after they occur. This reactive monitoring approach results in increased downtime, reduced reliability, and delayed response to system anomalies.

This paper proposes SentinelOps, an Artificial Intelligence based system monitoring platform designed to analyze application logs and predict potential system failures before they occur. The system integrates a web-based log generation environment, a machine learning powered log analysis engine, and a visualization dashboard for monitoring system health. Log data generated from the web application is processed using Python and Pandas, while machine learning algorithms such as Isolation Forest and Random Forest are used to detect anomalies and predict failure risk.

The processed insights are visualized through interactive dashboards that display system performance metrics, error frequency, and predicted failure probabilities. Experimental evaluation demonstrates that the proposed system can successfully detect abnormal behavior patterns in application logs and recommend corrective actions to administrators. SentinelOps provides an intelligent and proactive monitoring solution that enhances system reliability and reduces operational downtime.

Keywords - Artificial Intelligence, Log Analysis, Failure Prediction, System Monitoring, Machine Learning, Power BI Dashboard

I. Introduction

Modern digital services rely heavily on complex IT infrastructures consisting of web servers, databases, cloud services, and distributed applications. These systems continuously generate application logs, which record system activities such as request processing, authentication events, response times, error messages, and database transactions.

Application logs serve as a valuable source of information for diagnosing system issues and monitoring performance. However, large-scale systems generate thousands of log entries every minute, making manual monitoring extremely challenging.

Traditional monitoring systems depend on rule-based alert mechanisms that trigger notifications when predefined thresholds are exceeded. While such systems are useful, they are reactive in nature, meaning that they detect failures only after system breakdown occurs.

Reactive monitoring approaches lead to several issues including:

- Increased system downtime
- Delayed incident response
- Reduced system reliability
- Difficulty in identifying root causes of failures

To overcome these limitations, Artificial Intelligence and Machine Learning techniques can be applied to automatically analyze log data and identify abnormal patterns that indicate potential system failures.

This research introduces SentinelOps, an AI-based system monitoring platform designed to perform intelligent log analysis, anomaly detection, and failure prediction. The system collects logs from a web application environment, processes them using machine learning algorithms, and visualizes system analytics through a monitoring dashboard.

By combining machine learning techniques with data visualization tools, SentinelOps enables system administrators to monitor system health in real time and take preventive actions before failures occur.

II. Literature Survey

Several research studies have explored the use of log analysis techniques for system monitoring and anomaly detection.

Traditional log monitoring systems rely on manual inspection or rule-based detection methods, where predefined conditions trigger alerts. Although these systems are widely used, they lack the capability to detect complex patterns or predict future failures.

Recent research focuses on machine learning based monitoring systems that analyze large volumes of log data to identify abnormal patterns.

Xu et al. proposed a log mining approach that converts unstructured log data into structured event templates for anomaly detection. Their work demonstrated that machine learning algorithms can significantly improve failure detection accuracy.

He et al. introduced a deep learning based log analysis framework called DeepLog, which uses recurrent neural networks to model system behavior and detect anomalies in log sequences.

Similarly, Zhang et al. explored the use of clustering and classification algorithms for detecting abnormal system behavior from log files.

Despite these advancements, many existing systems still lack integrated monitoring platforms that combine log analysis, failure prediction, and visualization dashboards.

The proposed SentinelOps system addresses this gap by providing a complete AI-based monitoring solution that includes log collection, anomaly detection, failure prediction, solution recommendation, and data visualization.

III. Problem Statement

Modern IT infrastructures produce massive volumes of log data that contain critical information about system performance and operational events. However, analyzing this data manually is inefficient and time-consuming.

Traditional monitoring systems suffer from several limitations:

- Log analysis requires significant manual effort
- System failures are detected only after they occur
- Monitoring tools rely on static rule-based alerts
- Limited insights into root causes of system errors
- Lack of predictive monitoring capabilities

These challenges lead to increased system downtime and reduced operational efficiency.

Therefore, there is a need for an intelligent monitoring system capable of automatically analyzing log data, detecting anomalies, predicting failures, and providing actionable insights.

IV. Proposed System Architecture

The SentinelOps system is designed using a multi-layer architecture that integrates log generation, log processing, machine learning analysis, and data visualization components.

The architecture consists of four major layers:

1. Log Generation Layer

The web application generates logs during user interactions. These logs contain information such as request timestamps, HTTP status codes, response times, and error messages.

2. Log Processing Layer

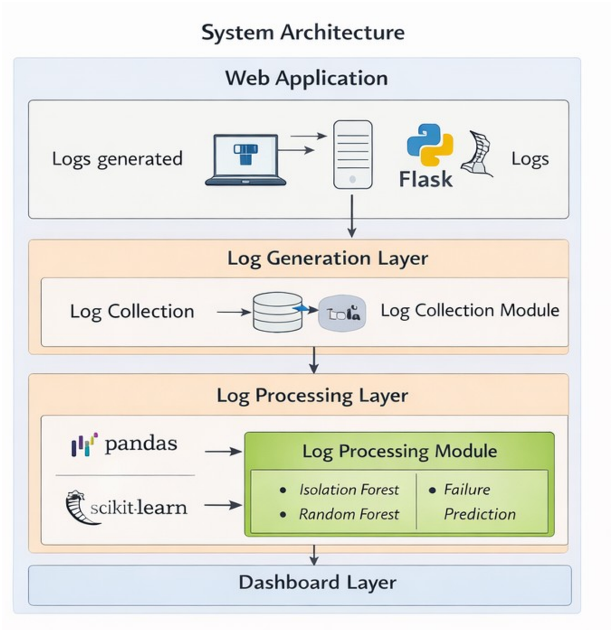
The collected logs are processed using Python and Pandas to clean, transform, and structure the data for analysis.

3. Failure Prediction Layer

Machine learning algorithms analyze the processed log data to detect anomalies and estimate system failure risk.

4. Visualization Layer

System analytics and monitoring insights are displayed through interactive dashboards that enable administrators to track system performance.



V. System Design

A. Module Description

Log Collection Module

This module collects logs generated by the web application. Each log entry includes information such as timestamp, request type, response time, and error messages.

Log Processing Module

Raw log data is cleaned and converted into structured datasets using Python libraries such as Pandas. This process includes data filtering, feature extraction, and transformation.

Failure Prediction Module

Machine learning algorithms analyze processed log data to detect abnormal system behavior. Two algorithms are used:

Isolation Forest
Detects anomalies by isolating abnormal data points.

Random Forest Classifier
Classifies system behavior as normal or abnormal based on extracted features.

Solution Recommendation Module

Once anomalies are detected, the system maps the detected issues to predefined corrective solutions such as:

Restart application services
Optimize database queries
Verify authentication services
Check network configurations

Monitoring Dashboard Module

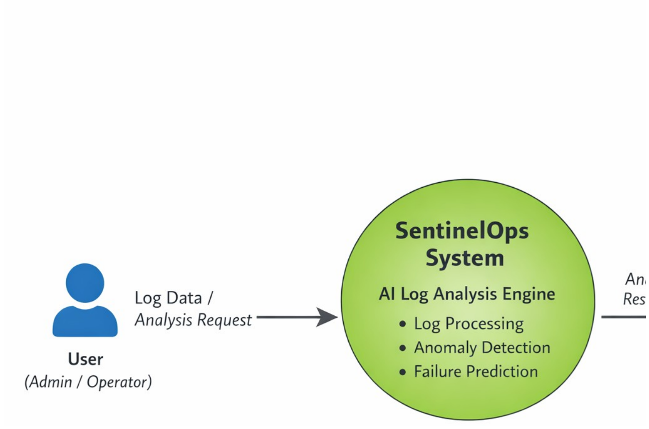
The monitoring dashboard provides real-time visualization of system metrics including:

Error frequency
Response time trends
Failure prediction results
System performance indicators

VI. Data Flow Diagram

Level 0 DFD

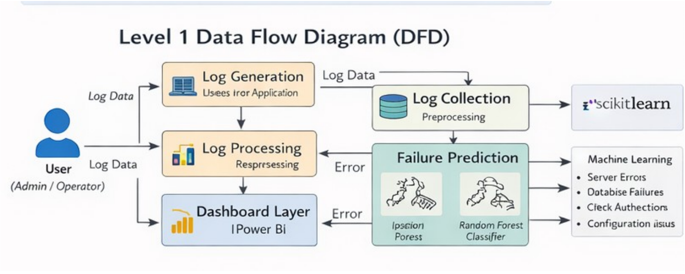
User → SentinelOps System → Monitoring Dashboard



The system collects logs, processes them using machine learning algorithms, and displays results through dashboards.

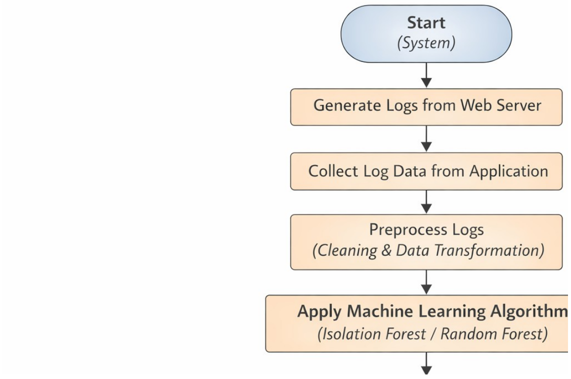
Level 1 DFD

Modules involved:



VII. Flowchart of the System

System workflow:


Start
(System)
Generate Logs from Web Server
Collect Log Data from Application
Preprocess Logs
(Cleaning & Data Transformation)
Apply Machine Learning Algorithm
(Isolation Forest / Random Forest)

VIII. System Implementatio

The SentinelOps system was implemented using Python and Flask framework.

The development environment included:

Python programming language
Flask web framework
Pandas data analysis library
Scikit-learn machine learning library
Power BI visualization tool

Log datasets were stored in CSV format and processed using machine learning algorithms to detect abnormal system behavior.

The system interface allows administrators to upload log files, analyze system performance, and visualize analytics through dashboards.

IX. Results and Discussion

Experimental evaluation was conducted using simulated log datasets containing normal system operations and various error conditions.

The results show that the SentinelOps system successfully detects several types of anomalies including:

Server errors
Database query failures
Authentication issues
High response time events

The machine learning algorithms effectively identify abnormal patterns in log data and estimate failure risk levels.

The Power BI dashboard provides visual insights into system performance, enabling administrators to quickly identify system issues and take corrective actions.

Overall, the system demonstrates improved monitoring capabilities compared to traditional rule-based monitoring systems.

X. Conclusion

This paper presented SentinelOps, an AI-based system monitoring platform designed to analyze application logs and predict system failures.

The system integrates machine learning algorithms with log analysis techniques to automatically detect anomalies and identify potential system failures.

By visualizing system analytics through interactive dashboards, SentinelOps enables administrators to monitor system health and respond to issues proactively.

The proposed system improves system reliability, reduces downtime, and enhances operational efficiency.

XI. Future Work

Future enhancements of the SentinelOps system may include:

Real-time log streaming using Apache Kafka
Integration with cloud monitoring platforms
Deep learning based anomaly detection models
Automated alert notifications via email or SMS
Integration with DevOps monitoring tools

References

[[1] F. T. Liu, K. M. Ting, and Z. H. Zhou,
“Isolation Forest,” Proceedings of the IEEE International Conference on Data Mining, 2008.

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

[3] Z. Chen, J. Liu, W. Gu, Y. Su, and M. Lyu,
“Deep Learning-Based System Log Analysis for Anomaly Detection,” IEEE Transactions on Dependable and Secure Computing, 2021.

[4] S. Ali, C. Boufaied, D. Bianculli, and L. Briand,
“A Comprehensive Study of Machine Learning Techniques for Log-Based Anomaly Detection,” 2023.

[5] M. Landauer et al.,
"Deep Learning for Anomaly Detection in Log Data: A Survey," Information and Software Technology, 2023.

[6] C. Hu, X. Sun, H. Dai, and H. Liu,
"Log Anomaly Detection Using Sentence-BERT and Bi-LSTM," Electronics Journal, 2023.

[7] Y. Duan et al.,
"LogEDL: Log Anomaly Detection via Evidential Deep Learning," Applied Sciences, 2024.

[8] Z. Ding,
"Anomaly Detection Approach Based on Isolation Forest for Streaming Data," IFAC Proceedings, 2013.

[9] J. Juknys,
"Anomaly Detection for System Logs: Literature Overview," 2023.

[10] H. Xu et al.,
"Deep Isolation Forest for Anomaly Detection," IEEE Transactions on Knowledge and Data Engineering, 2023.

[11] W. Li et al.,
"System Log Anomaly Detection Based on Contrastive Learning," Scientific Reports, 2025.

[12] S. Nedelkoski et al.,
"Self-Attentive Classification-Based Anomaly Detection in Unstructured Logs," 2020.