Supercharging Progress™

Log analyzer solution with human intelligence

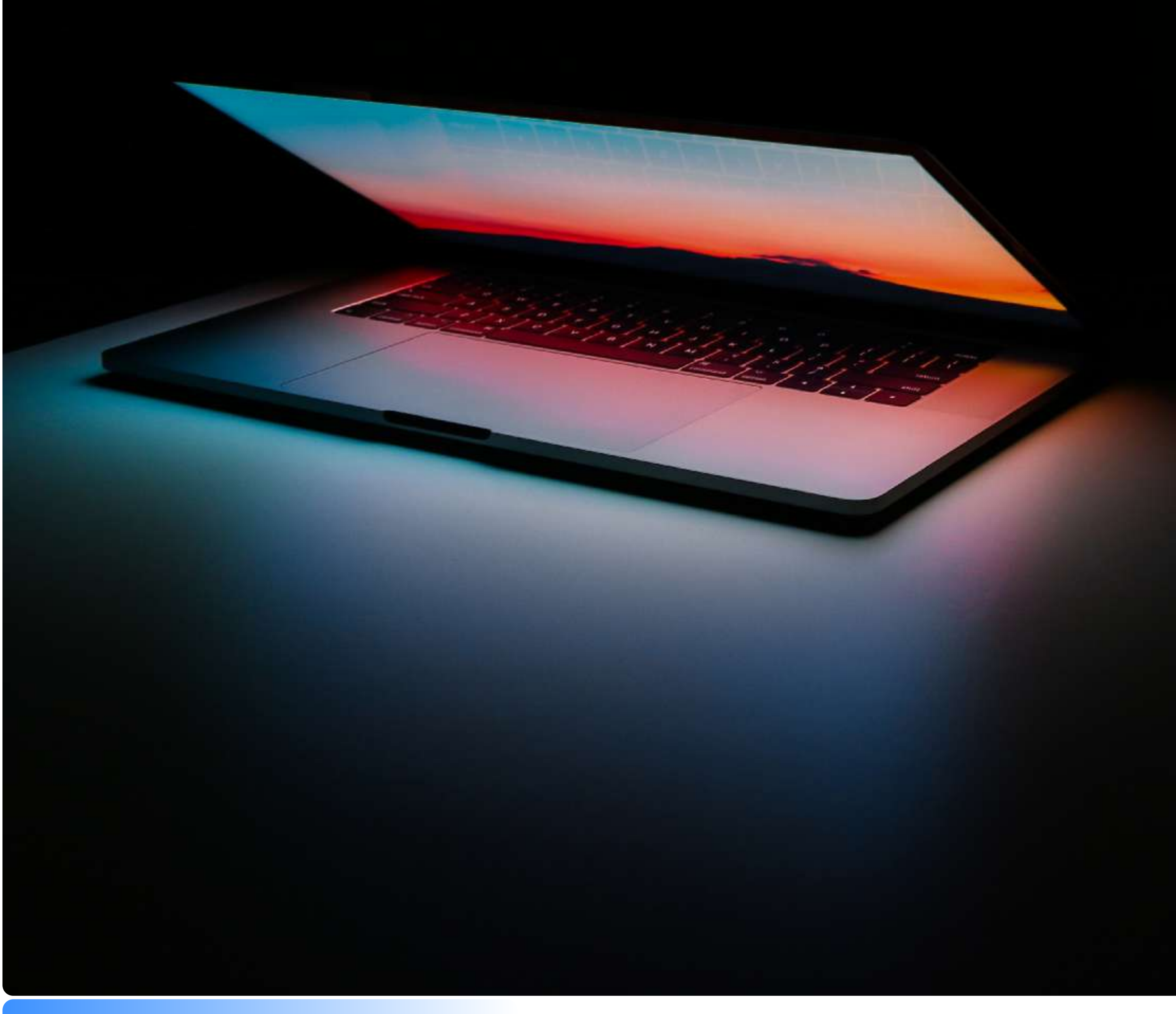


Table of contents

Abbreviations	3
Introduction	3
Business challenges	3
Problem statement	4
Solution	4
Benefits	9
Conclusion	9
References	10
Author information	11

Abbreviations

Short Form	Abbreviation
iLA	Intelligent log analyzer
AI	Artificial intelligence
RCA	Root cause analysis

Introduction

Software development is a vast field and developers face various challenges across all applications, irrespective of the domain and platform. Identifying and analyzing the cause of the issues during testing and deployment is a cost-intensive activity that requires lots of effort and time. To debug the issues, developers rely on the application logs for causal analysis. However, there are various challenges in analyzing the issues through logs, as given below:

- The log file size could be huge, making it difficult to analyze the issue in a short time.
- There could be scenarios where multiple log files need to be analyzed to identify the cause of the issue.
- There could be scenarios where the fix for past issues gets broken, resulting in the injection of similar issues. For such repeated issues, the same effort needs to be invested again in the analysis.

If there is a tool available that could learn from the historical data and help in identifying the anomalies and root cause of the issues, that would greatly improve the efficiency of analysis activity, thereby reducing the turnaround time.

In this paper, we will look at the Log Analyzer solution that learns from the available historical application log data and finds the anomalies and the root cause of the reported bugs in the software.

Business challenges

Developers rely on application logs to analyze the defects and find the root cause of the errors. However, there are several challenges faced during this analysis process:

- Manually analyzing the log files with huge volume data [more than 1 GB data] is time-consuming and challenging.
- Analyzing the application logs requires domain knowledge and a good understanding of the application flow. This leads to a dependency on SMEs for quick resolution.

- To address high-priority defects, quick turnaround time is critical.
- Triaging the defects to the right team is a challenge.

A log-based debugging tool with human intelligence can help in overcoming these challenges.

Problem statement

The developers rely on the application logs to analyze the defects and find the root cause of the errors. This log-based analysis requires domain knowledge and a good understanding of application flow. Furthermore, manual analysis of application logs takes a longer turnaround time to analyze and find the cause of the issue. These challenges impact the cost and schedule of the project.

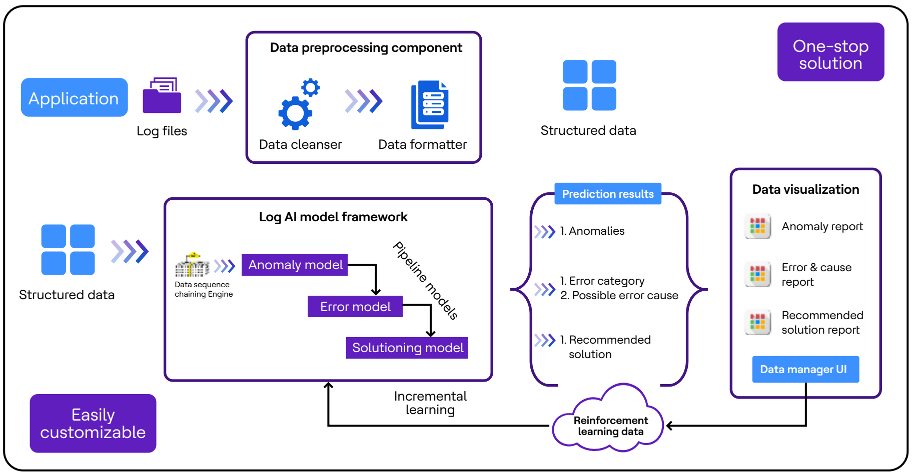
HCLTech has developed an AI-based solution called iLA (intelligent log analyzer) that leverages the historical log data of any application and acquires the knowledge gained from analyzing past hurdles into the learning model. This model can automatically analyze application logs while also finding anomalies and the cause of the problem. The human intelligence integrated into the model automates the analysis of defects and helps in reducing the turnaround time.

Solution

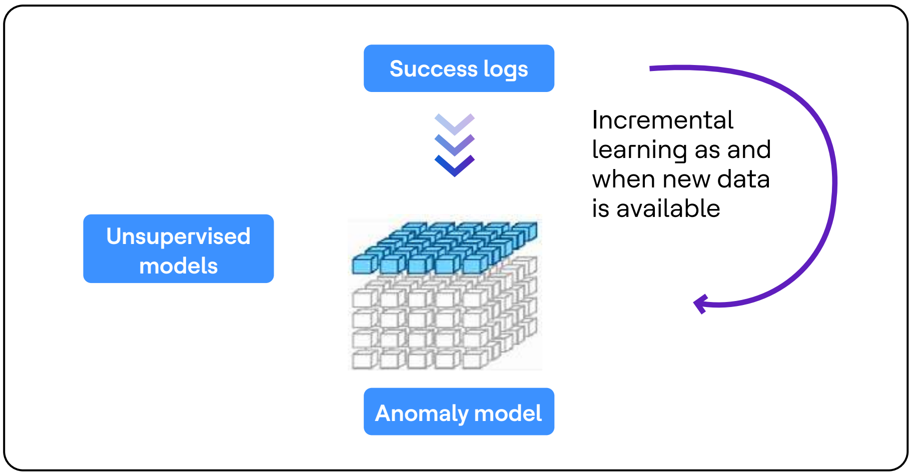
HCLTech's iLA tool enables the user to analyze and resolve errors with a quick turnaround time. The key features of this tool include easy defect triaging, which removes the hiccups in analyzing large-volume log files, enabling the identification of the anomalies in the log files and finding the flow sequence of the errors.

The primary goal of this tool is to minimize the dependency on the SME for defect analysis and find the root cause of the issue and its solution.

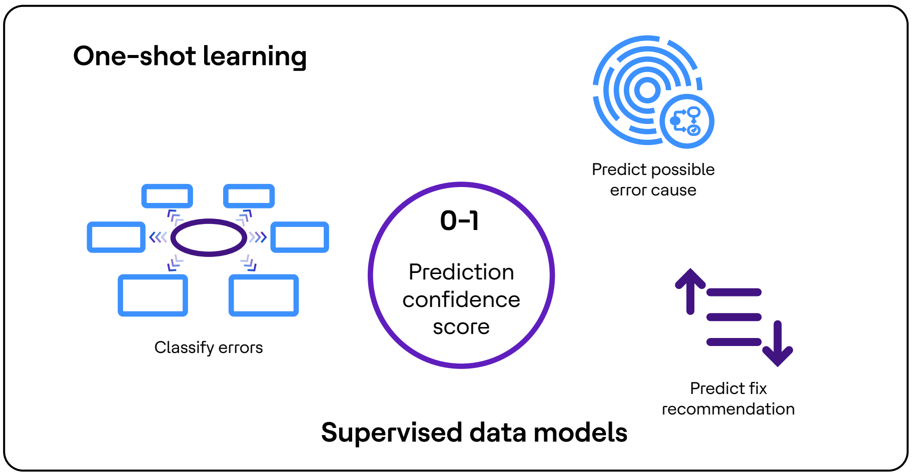
The SME's knowledge is imparted to the tool using artificial intelligence, feeding historical log data. The tool later uses this learning to remove the dependency on the SME for defect analysis. Thereby, iLA saves the SME's bandwidth as even the junior engineers can deliver the same results as the SME by using iLA. The diagram below depicts the iLA key components and workflow.



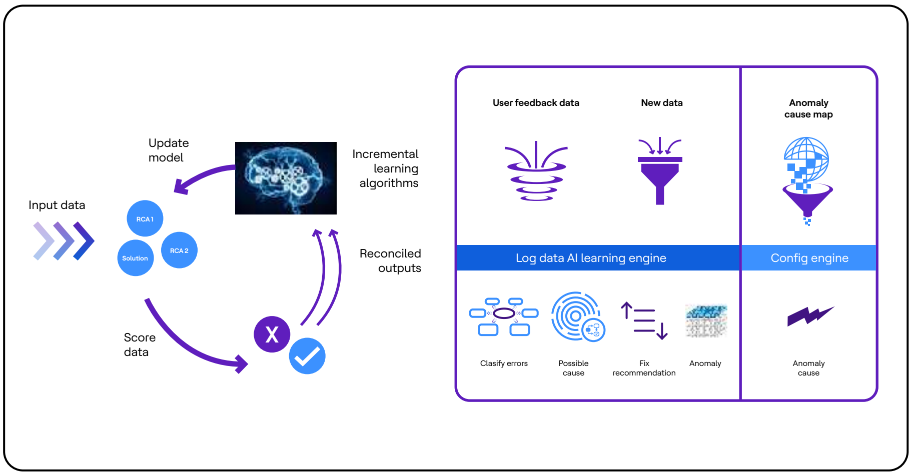
The iLA tool is built using both unsupervised and supervised data model frameworks. The unsupervised data model framework is built using the log files generated by all success scenarios. Here, the SME's knowledge is not required to build the intelligence to identify the anomalies. The system uses a deep learning algorithm to self-learn all the success scenarios and subsequently identify and report the anomalies and the flow sequence of errors.



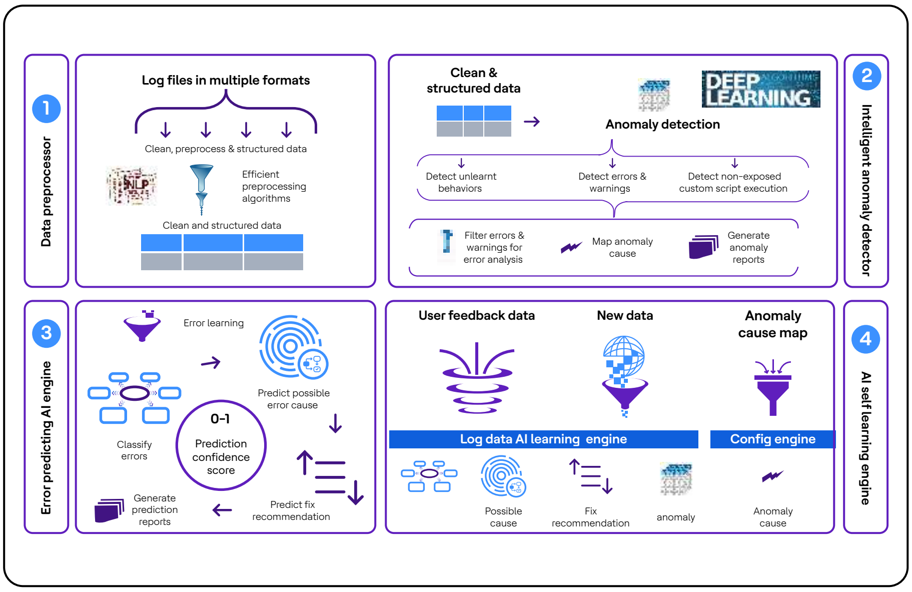
Identified anomalies and the flow sequence of the errors are further fed into the iLA-supervised data model framework. The supervised data model helps classify the error component, predicting the root cause of the error and the recommended solution. The supervised data model is built using the SME's knowledge. The knowledge gained is fed into the system by labeling the data and the model is trained using machine learning algorithms.



iLA's self-learning framework operates on learning through feedback. The tool incrementally learns through user feedback. It predicts error causes and recommendations, along with providing a confidence score. The confidence score helps the end user to either reject or accept the prediction. A confidence score of 1 confirms the prediction to be 100% accurate. A confidence score of < .5 requires the end user to further investigate the issue and update the correct data as feedback to the tool. The tool feeds the feedback to the supervised model framework for self-learning, which helps in incremental learning.



iLA prediction framework component architecture

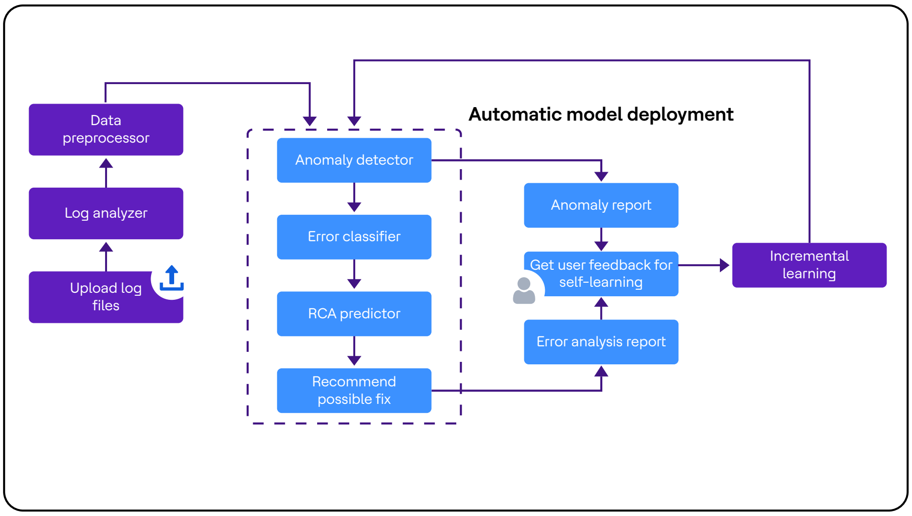


The prediction framework is comprised of four primary components:

- Data preprocessor: There could be various log files having multiple data formats. Further, the data could be unstructured and needs to be cleansed to bring structure. This component is responsible for data cleanup and feature engineering.
- Anomaly detection: This module detects the unlearned behavior in the software execution flow. It detects the errors and warnings for error analysis.
- Error-predicting AI engine: This module is based on the error learning. It starts with classifying the errors in the log files, predicting the possible error cause and the fix recommendation along with the confidence score. The confidence score helps in confirming the accuracy of the prediction.

- AI self-learning engine: When the confidence score of the prediction is less than 0.5, the SME is notified for further analysis and requested to share the results as feedback to the system. The SME's feedback is self-learned by the tool and the auto-upgraded model is auto-deployed to use the learnings for future predictions with higher accuracy.

iLA – execution workflow



- The iLA tool takes the log files as input. The user uploads the application log files and the data preprocessing cleanses the logs file data structures it and feeds it into the anomaly detector.

- The anomaly model detects the anomalies in the log files and reports them.

- The reported anomalies can be further mapped to a cause by the SME user for future inferences.

- The identified anomalies are filtered and all the errors and exception patterns are identified and fed into the error classifier. The error classifier model predicts the application component that caused the error.

- The predicted results help in triaging the defect to the right team. This helps avoid the challenges of assigning the right tickets to the right team.

- The log data, along with the classified error, is further fed into the RCA model, which, in turn, predicts the possible root cause along with the confidence score. The confidence score helps in confirming the accuracy of the predicted root cause.

- As a next step, the data, along with the predicted root cause, is fed into the recommendation model engine, which predicts the recommended fix for the reported error with the possible predicted root cause. This fix is also tagged with a confidence score.

- After the completion of the prediction process, the reporting engine of the iLA tool reports the prediction results of all the errors and exceptions, along with the confidence score.

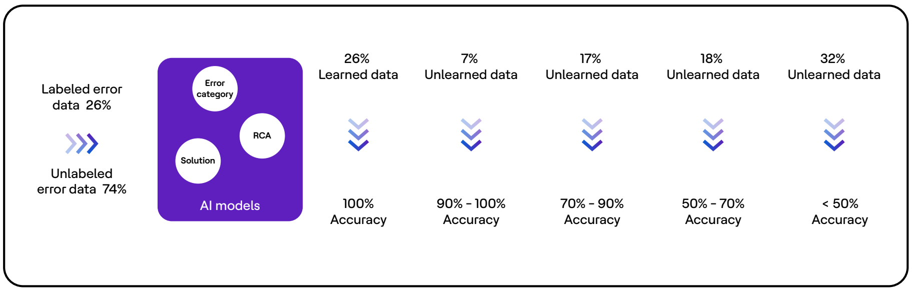
- The SME user can validate the prediction results where the confidence score is less than one and provide feedback.

- The tool learns from the feedback, models are auto-deployed and the learning is used for future prediction.

iLA case study

Training data [Labeled data] Error category	Test data	Prediction accuracy range and count										
		RCA 1										
		100%	90% - 99%	80% - 89%	70% - 79%	60% - 69%	50% - 59%	40% - 49%	30% - 39%	20% - 29%	10% - 19%	1% - 9%
56	1546	48	7	5	1	13	32	79	220	549	474	118

Training data determines the application's defect analysis coverage and prediction accuracy. We identified a small volume of data for training and used a one-shot algorithm with Siamese Network to train the model. Subsequently, this model was tested with data that was part of the training dataset and data that was not. The greatest achievement is the highest accuracy of 100% on the learned data achieved with a very low volume of training samples.



The table below provides a comparative analysis of the Log Analyzer with the traditional log analysis process:

Ticket type	Tradition log analysis process		iLogAnalyzer tool	
	Turnaround time	Engineer productivity	Targeted turnaround time	Targeted engineer productivity
Complex (P1, P2)	5 days	1 / week	2 .5 days	2 / week
Medium(P3)	2.5 days	2 / week	1 day	5 / week

Benefits

Through the usage of the iLA tool, the following benefits could be achieved, as elaborated in this paper:

- The key specialization of the tool is the application logs-based learning. It gives highly reliable results with the prediction of 98% accuracy on the learned data.
- Facilitated a 50% reduction in turnaround time in defect fixing.
- Defect fix productivity was increased by 50%.

Conclusion

With its human intelligence, iLA helps in analyzing the log files to find anomalies and identifies the cause of the error along with a possible solution. The automatic analysis with high accuracy helps in quick turnaround time in defect analysis and fixing.

The self-learning capability of the tool through user feedback helps in maintaining the consistency of the prediction.

iLA can be easily customized for any application and deployed with minimal effort. The tool can even be used by less experienced resources and contribute equally to the SME's productivity.

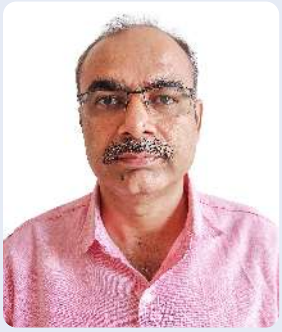
References

- https://paperswithcode.com/paper/prototypical-networks-for-few-shot-learning
- https://iq.opengenus.org/one-shot-learning-in-ml/
- https://www.keycdn.com/blog/log-analysis-tools
- https://logparser.readthedocs.io/en/latest/tools/Drain.html
- https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Author information

Sasikala Sethu

Sasikala Sethu is working as an Assistant General Manager with PTS Solutions CoE. She has vast experience in Microsoft technologies, data science and analytics. Along with a passion for training, she has more than 24 years of experience in IT with expertise in providing solutions to any given problem statement and architecting the solution with loosely coupled design.



Abhishek Tomar

Abhishek Tomar is working as Associate General Manager with HCLTech. He is leading the PTS Solutions CoE Professional Services team and has vast experience in software development and testing for embedded systems in the consumer electronics domain. His interest lies in exploring machine learning methods to solve industry problems.

About HCLTech

HCLTech is a global technology company, home to more than 223,400 people across 60 countries, delivering industry-leading capabilities centered around digital, engineering, cloud and AI, powered by a broad portfolio of technology services and products. We work with clients across all major verticals, providing industry solutions for Financial Services, Manufacturing, Life Sciences and Healthcare, Technology and Services, Telecom and Media, Retail and CPG and Public Services. Consolidated revenues as of 12 months ending June 2023 totaled $12.8 billion. To learn how we can supercharge progress for you, visit hcltech.com.

HCLTech | Supercharging Progress™

hcltech.com