# ABSTRACT

Anomaly detection is an important aspect of managing large scale file systems that are sourced from distributed systems. Log monitoring has become a key area of research and evaluating them as they are being recorded leads to actionable insights. Log messages are now broadly used in cloud and software systems. With the increased scale and complexity of the data coming in as logs, manual inspection of them seems infeasible. There are several detection methods based on automated log analysis. However, the area of anomaly detection in the context of machine learning models seems relatively untouched. Our approach aims to perform dimensionality reduction and feature extraction from the templates obtained from the logs and deploy machine learning models to evaluate their performance using cross validation techniques. This paper gives a detailed approach of how we use an unstructured HDFS log file through a LogParser library that uses an unsupervised learning algorithm to generate structured log files as an output. The event templates are labelled based on BlockIds that make up a trace. We then perform feature extraction on the unique log entries to consider only the important dimensions for the features that will further help narrow down detection of anomalies. We then try numerous classification models and compare the metrics to understand how the anomalies occur and the key features that are weighted in predicting if an incoming log file is an anomaly.

# INTRODUCTION

Logs are widely used to record machine runtime information, such as timestamps, ID, levels and log components, and the current state of the log entry. Gaining visibility into modern IT and software environments is a challenge that a number of organizations are finding difficult to overcome. In today’s growing digital world, almost all systems leave a digital footprint of their operational status, configurations, environmental changes and errors into an event log of a particular template. This gives us an excellent chance to understand the operational status of the systems, health and performance of the internal programs, computing infrastructure, network and security analysis of downtime incidents.

Data centers consist of thousands of software components that report a million misbehaves. Traditionally developers use print commands and complex monitoring libraries to record trace execution and runtime statistics. Large scale services use multiple log parser libraries[1] that produce structured information from the logs coming from various sources. Typically, a log message records a specific system event with a set of fields: *timestamp(*recording the occurrence of the event), severity level, message content, template and components. An example of the logs is shown below -

\*\*\*Example\*\*

As seen from the example, the raw message can be split into two parts: constant part and the variable part. This is classified as the event template for different log events in the dataset. For instance, The sentence “ I am going to the mall” will be parsed into the template “ I am going to the <\*>” with “mall” as the parameter.

[1]An Evaluation Study on Log Parsing and Its Use in Log Mining(https://pinjiahe.github.io/papers/DSN16.pdf)