Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

# Advancing Log Analysis with Synthetic Data: Techniques, Application, and Challenges

**[*1]Sonali Vidhate , [2]Dr.Pankaj Dashore**

[1]Department of Computer Science and Engineering,Research Scholar at Sandip University and Assistant Prof. at MET's Bhujbal Knowledge City,Nashik Maharashtra

[1]*sonali.vidhate878@gmail.com*

[2]Department of Computer Science and Engineering,Sandip University,Nashik Maharashtra

[2]*dashorepankaj@gmail.com*

## ABSTRACT

Log data is an essential resource for system monitoring, diagnosis, and optimization in a variety of disciplines in today's data-driven contexts. In order to establish a cohesive framework for improved analysis and decision-making, this study explores the integration of log data from several systems. The goal of this paper is to address the difficulties posed by the amount, complexity, and heterogeneity of log data by utilizing cutting-edge methods in machine learning, data standardization, and log parsing. By combining log data from several sources, the suggested structure allows for centralized analysis and storage while preserving context and data quality. The results demonstrate how integrated log data can improve system resilience and predictive analytics in addition to operational efficiency. This study will help to collect logs from a distributed system, which will provide us with a comprehensive insight of the system's behavior and performance. Because the logs contain a wide range of activities, including system events, errors, and performance indicators, they offer valuable information for understanding and enhancing distributed systems.

**Keywords:** Log parsing, Performance indicators, System resilience

## INTRODUCTION

Log data has emerged as a crucial element for tracking, evaluating, and improving system performance in the age of digital transformation. Applications, servers, networks, and devices all produce logs, which offer a multitude of data regarding system functions, user interactions, and possible irregularities. Nonetheless, the increasing intricacy of contemporary systems leads to a varied and extensive compilation of log data, frequently emanating from numerous, disparate sources. Comprehensive analysis and decision-making are severely hampered by the

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

fragmentation. Consolidating log data from multiple sources into a single framework that facilitates easy analysis and useful insights is known as integrated log data. Issues including data heterogeneity, inconsistent log formats, and the requirement for real-time processing to remain relevant in dynamic situations are all addressed during the integration process. Even though traditional log analysis tools offer useful features, they frequently have limitations when working with large-scale log systems that have several sources. Therefore log data is essential for monitoring, evaluating, and improving system performance across various domains, including applications, servers, networks, and devices. It provides critical insights into system functions, user interactions, and potential irregularities [1].

By utilizing developments in data engineering and machine learning, this study attempts to investigate approaches and frameworks for efficiently integrating log data. This paper aims to facilitate enhanced system monitoring, quicker anomaly detection, and better predictive capabilities for a variety of applications, such as industrial systems, cybersecurity, and IT infrastructure management, by establishing a single repository for log data. Log data integration and analysis have been extensively studied in a number of fields, such as industrial systems, cybersecurity, and IT operations. The issues of log data heterogeneity, scalability, and real-time processing are the main focus of current study. This study fills important gaps in scalability, heterogeneity, and actionable insights while building on the underlying work in log processing, integration frameworks, and advanced analytics[1]. The goal of this research is to advance integrated log data systems by fusing cutting-edge approaches with creative strategies.The advancements in data integration frameworks, log processing, and sophisticated analytical methods. Logs are records of events or processes within a system, application, or network. They provide vital insights into System performance (e.g., CPU usage, memory allocation) , User activity (e.g., login attempts, actions), Error tracking (e.g., exceptions, crashes), Security monitoring (e.g., access logs, threat detection)[1][2].

Unlike real logs, which are generated by actual operations within a system, synthetic log data is intentionally created using predefined rules, machine learning models, or stochastic processes. It is very important because it plays a crucial role in scenarios where real-world logs may be unavailable in new systems or applications that might not yet produce sufficient logs. The sensitive logs from production environments often contain sensitive information, limiting their use for development or testing. Collecting and storing large volumes of real logs can be expensive. It is incomplete in real logs, may lack diversity or fail to represent edge cases.The primary goal of the research is to enhance system monitoring, accelerate anomaly detection, and improve predictive capabilities across various applications, including industrial systems, cybersecurity, and IT infrastructure management. This is achieved by establishing a single repository for log data.

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

## CONTRIBUTION

In this paper, we will collect logs from a distributed synthetic log dataset, which will provide us with a comprehensive insight of the system's behavior and performance. Because the logs contain a wide range of activities, including system events, errors, and performance indicators, they offer valuable information for understanding and enhancing distributed system architectures.It also records the date and time of each event in the format [2023-11-20T08:40:50.664842], which provides a chronological sequence for system operations. Log level shows the severity or importance of the recorded event and gives information about the importance of system events by classifying entries into levels such as INFO, WARNING, ERROR, or FATAL. By giving the name or identification of the service associated with each log entry, it makes it possible to classify and analyze events based on the modular components of the distributed system. By giving detailed information on the logged event, the message sheds light on the type and context of the distributed system activity. Because each request is individually recognized by its RequestID, log entries connected to specific processes or transactions in the distributed system may be tracked down and correlated. By providing information about the entity interacting with the distributed log data and reflecting the user associated with the logged event, users aid in user-centric analysis. Tracking and analyzing the activities of different clients in the distributed system is made simpler by ClientIP.

## ALGORITHM

To create a unified and scalable indexing mechanism for heterogeneous log data, ensuring efficient querying, retrieval, and analysis. For Input two parameters are required as collection of log entries and metadata.

*Input:*

L={ l1, l2, ..., ln } ;collection of log entries

L = {l_1, l_2 , ..., l_n }

M = { m1, m2, ..., mk } ;Metadata (e.g., timestamps, source IDs).

M = {m_1, m_2, ..., m_k}

**Output:** A centralized index for fast and efficient retrieval of logs.

**Steps**

The defining of input parameters, which establishes the log generation's goal and specifies log structure, volume, and behavioral patterns, is the first step in the flow of synthetic log data. The data modeling step comes next, during which log formats and templates are created and pre-established rules or algorithms are used to mimic real-world behaviors, trends, or anomalies.

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

After the models are established, the log creation procedure is carried out, using randomized simulations, rule-based approaches, or AI/ML techniques to produce artificial logs that closely resemble actual situations. To make sure the generated logs satisfy structural and behavioral requirements, such as volume compliance, pattern accuracy, and schema validation, they go through validation and quality checks.

Following validation, the synthetic logs are kept and handled in databases, data lakes, or log management platforms. To facilitate effective retrieval, they are indexed and categorized according to time, source, or severity. After that, these logs are used for a number of tasks, including anomaly detection, machine learning training, performance simulation, and software testing. The log production process is improved to increase realism and adjust to changing requirements using the insights and outcomes from these operations, which input into a feedback loop. High-quality synthetic log data is created and used efficiently for a variety of applications thanks to this iterative system cycle.

*1. Log Preprocessing:*

Log Parsing: For each log entry extract structured fields using a parsing algorithm (e.g., Drain or Spell).Example fields: Timestamp, Source, Event Type, Severity Level, Message.

Normalization: Convert extracted fields into a standardized format. Example: Convert all timestamps to ISO 8601 format.

*2.Metadata Augmentation:*

Enhance log entries with additional metadata like Host information, application name, or geographic location. Example: Append log source identifiers (e.g., IP addresses, server IDs).

*3.Index Creation:*

Define Indexing Fields Choose key fields for indexing, such as: Timestamp (for temporal queries), Event Type (to group similar events), Source ID (to identify log origins), Create a compound index for multi-dimensional queries, Tokenization and Keyword Extraction, To enable full-text search, tokenize log messages into keywords, Use techniques like TF-IDF to prioritize significant terms.

*4.Mechanism of indexing*

*Tree-Based Indexing:* Use of B+ tree or R-tree for indexing structured fields like timestamps and source IDs.

*Inverted Indexing:* Build an inverted index for unstructured fields like log messages. Example: Map keywords to log entry IDs for fast full-text search.

*Hierarchical Indexing:* Create hierarchical indices for logs with nested fields (e.g., JSON logs).

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

*5.Index Storage:* Store the indices in a scalable database, such as Elasticsearch, that supports distributed and real-time indexing. Partition the index based on key fields (e.g., time or log source) to optimize query performance.

*6.Query Optimization*

*Query Parsing:* Parse user queries to identify relevant fields and keywords.

*Index Traversal:* Retrieve matching entries using the appropriate index structure.

*Ranking and Filtering:* Rank results based on relevance using a scoring mechanism (e.g., BM25). Apply filters for additional constraints (e.g., severity or specific time range).

*7. Index Maintenance*

Periodically update indices to account for new logs or changes in the dataset. Optimize and merge indices to avoid fragmentation.

```
def create_index(logs, metadata):
    index = {}
    for log in logs:
            parsed_log = parse_log(log)
            normalized_log = normalize_fields(parsed_log)
            enriched_log = augment_metadata(normalized_log, metadata)
      for field, value in enriched_log.items():
          if field not in index:
              index[field] = {}
          if value not in index[field]:
              index[field][value] = []
          index[field][value].append(log['id'])
    return index
```

The algorithm is implemented in the create_index function in the systematic way to generate a centralized and efficient index for querying log data. To create a unified index from a collection of log entries, enrich them with metadata and build an index that allows efficient lookups for specific field values across the logs. This index organizes the logs for efficient searching and retrieval based on specific fields (e.g., timestamps, severity, or source).

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

## PERFORMANCE EVALUATION AND EXPERIMENTAL SETUP

The Distributed System generates a wide range of events and anomalies across numerous components, nodes, and workloads in order to replicate the complexity of an actual distributed environment. This structured and flexible data format can be used to test, train, and evaluate algorithms for tasks such as anomaly detection, log parsing, root cause analysis, and distributed system monitoring.

The Synthetic Log Data of Distributed System dataset is a widely used tool for modeling the behavior, fault scenarios, and performance of a distributed computing environment. This type of dataset is typically used to train and assess algorithms in fields such as anomaly detection, log parsing, and distributed system monitoring. Log data must be structured to represent different performance measures across time or events in order to be turned into a performance matrix. An example of such a matrix for a system's performance is provided here, along with some data fields are as follows:

**Table 1** ***System's Performance***

| Timestamp | Service Name | Response Time (ms) | CPU Usage (%) | Memory Usage (MB) | Error Rate (%) | Throughput (req/sec) |
|---|---|---|---|---|---|---|
| 10-11-2023 10:00 | Service A | 120 | 70 | 512 | 0.5 | 200 |
| 10-11-2023 10:05 | Service B | 98 | 65 | 450 | 0.3 | 220 |
| 10-11-2023 10:10 | Service C | 105 | 70 | 530 | 0.1 | 250 |
| 10-11-2023 10:15 | Service A | 130 | 80 | 540 | 0.2 | 210 |
| 10-11-2023 10:20 | Service B | 95 | 60 | 460 | 0 | 230 |

The following columns in the table display important performance indicators over time for various services in a distributed system:

*Timestamp:* The date and time of the logged metric. This could be in intervals, like every 5 minutes, hourly, or based on each transaction.

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

*Service Name:* The name of the service or component within the system (e.g., Service A, Service B).

*Response Time (ms):* Time taken for the service to respond to a request, often a key performance metric.

*CPU Usage (%):* Percentage of CPU resources used by the service, indicating the workload.

*Memory Usage (MB):* Amount of memory consumed, helping gauge memory efficiency.

*Error Rate (%):* Percentage of requests that resulted in errors, helping to monitor service reliability.

*Throughput (req/sec):* Requests per second the service handles, indicating load capacity.

**IMPLEMENTATION**

The purpose of this study is to create and examine synthetic log data. The timestamp, log level (INFO, ERROR, etc.), and custom messages are used to create synthetic logs.Next, highlight abnormalities (such as surges in ERROR logs) by aggregating data and counting instances of each log level or event type over time.For instance, 24-hour timestamps, genuinely dispersed log levels, and anomalies that are shown as spikes in ERROR or CRITICAL logs. The frequency of various log levels over time is displayed in the following synthetic log data visualization:

*1. Normal Patterns:* As may be expected given their higher probability, INFO logs predominate, followed by DEBUG, WARNING, and other logs.

*2. Anomaly Spike:* Around 15:00, there was a noticeable spike in ERROR logs, which suggested a system problem.

The goal of study shows the frequency of various log levels over time, such as INFO, ERROR, and DEBUG. An anomaly or system problem is indicated by a spike of ERROR logs at a certain period, while INFO logs predominate during regular operation, according to the insight of log level frequency over time. It tracks the distribution of different log levels (e.g., INFO, DEBUG, WARNING, ERROR) over time. INFO logs dominate during normal system operations, reflecting routine events. Spikes in ERROR logs at specific times indicate system anomalies or failures. DEBUG and WARNING logs appear intermittently, showing intermediate troubleshooting or cautionary events. Below graph helps to detect trends and pinpoint anomalies in system behavior, aiding in quick diagnostics.

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

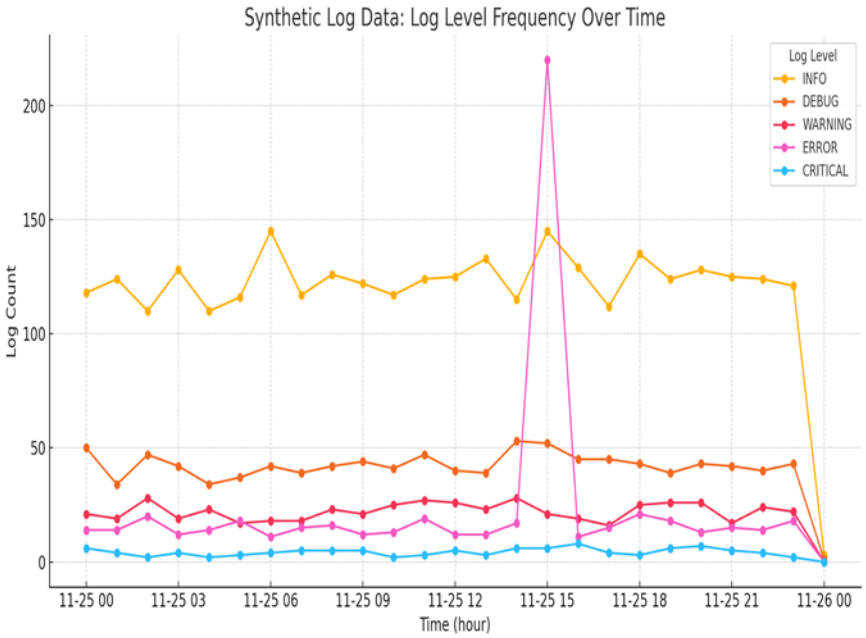


**Figure 1** ***Chart Representation of log level frequency over Time***

Graph shows the frequency of different log levels such as INFO, DEBUG, WARNING,ERROR,CRITICAL etc. and mainly four log levels are mentioned in data which are INFO, ERROR, DEBUG, WARNING. A synthetic log file typically looks like in the form of timestamp, loglevel, Messages and additional fields are there. Log levels are recorded in the current timestamp and according to that user_id,transaction_id and memory usages are created. The different messages are generated as per log level and log messages are scattered in various nodes. The structure and diversity of synthetic log data, capturing various events and system activities. A sample of synthetic log files include Timestamp, Log Level, Message and additional fields.

**Table.2** ***Synthetic log file***

| **Timestamp** | **Log Level** | **Message** | **Additional Fields** |
|---|---|---|---|
| 25-11-2024 15:42 | INFO | User 234 logged in | user_id: 234, ip: 192.168.1.45 |
| 25-11-2024 15:42 | ERROR | Transaction failed: Insufficient balance | transaction_id: 453, balance: 0.0 |

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

| | | | |
|---|---|---|---|
| 25-11-2024 15:42 | DEBUG | Memory usage at 78% | memory_usage: 1.5GB |
| 25-11-2024 15:42 | WARNING | Unusual login attempt detected | ip: 203.0.113.10 |

The visualization could be represented in log frequency by Level and Log event trends over time are displayed in line charts. The distribution of Log Levels over the frequency of INFO, DEBUG, ERROR, and so on. Also anomaly Detection highlights unusual spikes in logs. A dominant presence of INFO and DEBUG logs indicates normal operations. A Synthetic Log Data of Distributed System outcome analysis highlights important findings regarding performance, anomaly patterns, and system interdependence. By closely analyzing log data, it finds bottlenecks, improves resource allocation, and makes the system more resilient to mistakes and failures.

**Table 3** ***Performance matrix of comparison of message***

**Message**

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | Algorithm Steps | 6479 | 6.5 | 6.5 | 6.5 |
| | Crashes | 5968 | 6.0 | 6.0 | 12.4 |
| | Critical Errors | 7842 | 7.8 | 7.8 | 20.3 |
| | Data Corruption | 5957 | 6.0 | 6.0 | 26.2 |
| | Database Errors | 8009 | 8.0 | 8.0 | 34.3 |
| | File I/O | 6613 | 6.6 | 6.6 | 40.9 |
| | Input Validation Warnings | 8622 | 8.6 | 8.6 | 49.5 |
| | Performance Warnings | 8774 | 8.8 | 8.8 | 58.3 |
| | Resource Warnings | 8785 | 8.8 | 8.8 | 67.0 |
| | SQL Queries | 6267 | 6.3 | 6.3 | 73.3 |
| | Startup Messages | 10048 | 10.0 | 10.0 | 83.4 |
| | Status Updates | 10094 | 10.1 | 10.1 | 93.5 |
| | Trace Information | 6542 | 6.5 | 6.5 | 100.0 |
| | Total | 100000 | 100.0 | 100.0 | |

Table 3 Summarizes the frequency, percentage, and cumulative percentage of different messages (e.g., algorithm steps, warnings, errors).

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

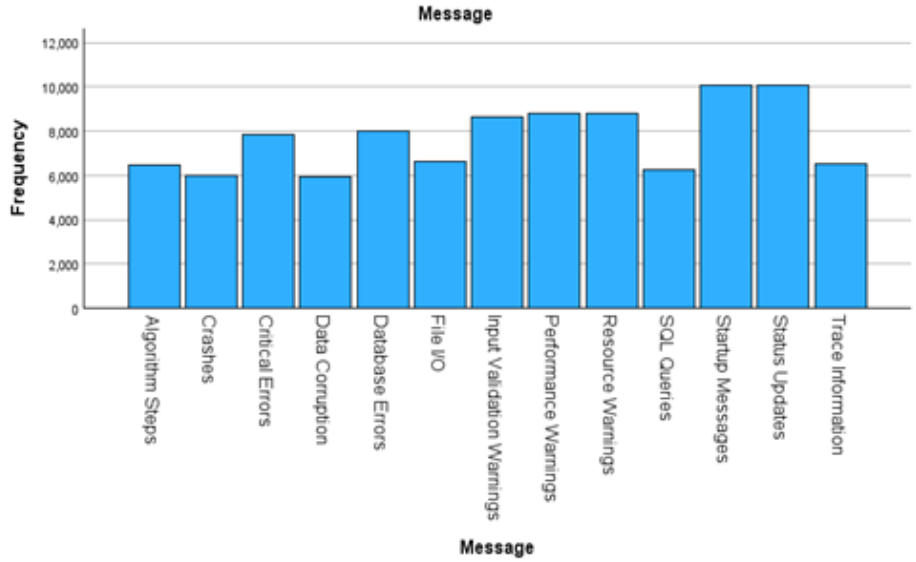


**Figure 4** ***Graphical representation of performance comparison of message***

It displays the frequency of different log levels and its purpose is to understand the distribution of log levels across the dataset. A high count of INFO logs suggests normal operations, while a spike in ERROR or CRITICAL logs may indicate system issues. There are one million in all. LogLevel appears when handling data and displays messages such as DEBUG, ERROR, FATAL, INFO, WARNING, and so on.The log level of events on the data is used to construct the graph.Frequency, percentage, valid percent, and cumulative percent are determined.

**Table 4** ***Performance matrix of comparison of log level***

**LogLevel**

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | DEBUG | 25901 | 25.9 | 25.9 | 25.9 |
| | ERROR | 15851 | 15.9 | 15.9 | 41.8 |
| | FATAL | 11925 | 11.9 | 11.9 | 53.7 |
| | INFO | 20142 | 20.1 | 20.1 | 73.8 |
| | WARNING | 26181 | 26.2 | 26.2 | 100.0 |
| | Total | 100000 | 100.0 | 100.0 | |

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

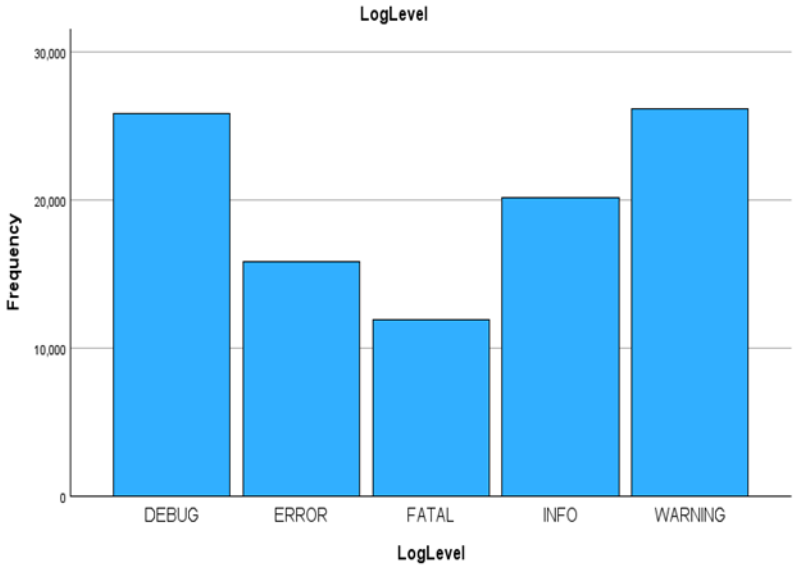


**Figure 5** ***Graphical representation of performance comparison of log level***

A total of one million services are available for processing data.Data uses ServiceA, ServiceB, ServiceC, and ServiceD.The services that take place on the data are the basis for creating the graph.Frequency, percentage, valid percent, and cumulative percent are determined. A performance matrix is a tabular or graphical representation that compares various services or components of a system based on key performance indicators (KPIs) derived from synthetic log data. It helps analyze and benchmark the efficiency, reliability, and scalability of services in a simulated environment.

**Table 5** ***Performance matrix of comparison of services***

**Service**

| | | Frequency | Percent | Valid Percent | Cumulative Percent |
|---|---|---|---|---|---|
| Valid | ServiceA | 24886 | 24.9 | 24.9 | 24.9 |
| | ServiceB | 24775 | 24.8 | 24.8 | 49.7 |
| | ServiceC | 25200 | 25.2 | 25.2 | 74.9 |
| | ServiceD | 25139 | 25.1 | 25.1 | 100.0 |
| | Total | 100000 | 100.0 | 100.0 | |

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

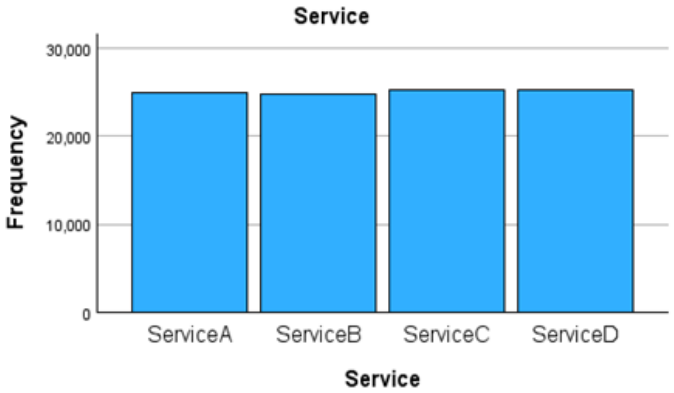


**Figure 6** ***Graphical representation of performance comparison of services***

A total of one million services and messages are included in data handling.Performance warnings, critical errors, algorithm steps, crashes, file I/O, input validation warnings, resource warnings, SQL queries, startup messages, status updates, trace information, and database failures are among the four services—Services A, B, C, and D—that are utilized in data and messages.The services and message occurrences on the data are used to construct the graph.Frequency, percentage, valid percent, and cumulative percent are determined.

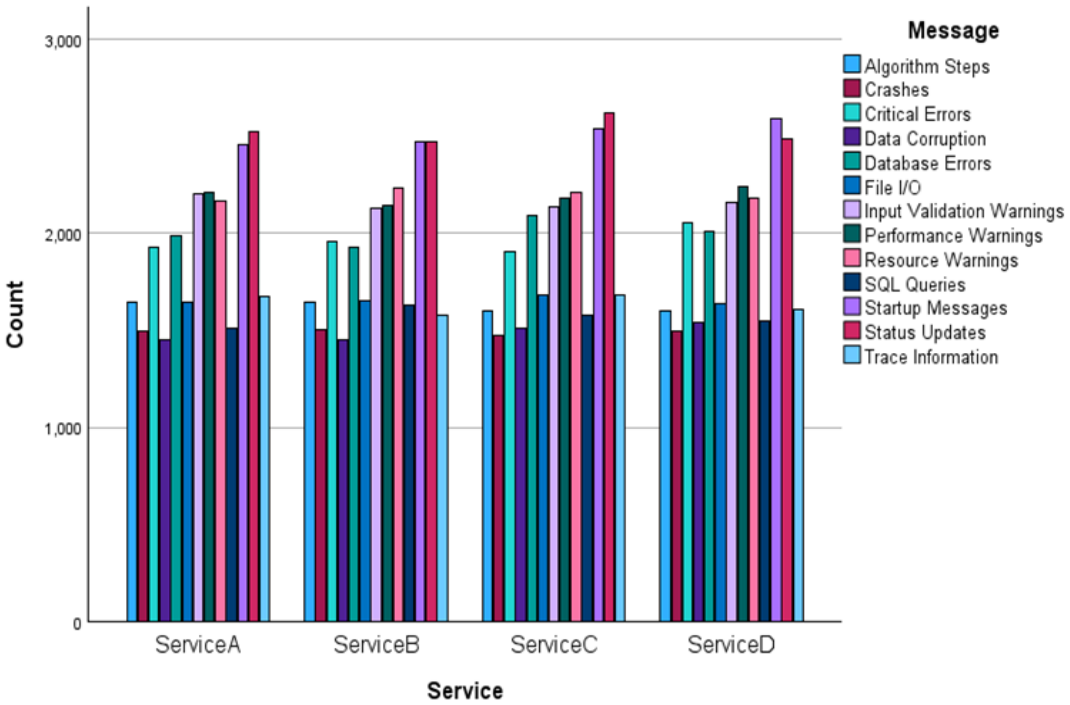


**Figure 7** ***Graphical representation of performance comparison of services via message***

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

## CONCLUSION

Synthetic log data refers to artificially generated log entries, typically used for testing, training, and simulating environments where real-world data may not be available or appropriate. The generation of synthetic log data helps to model complex systems, simulate various scenarios, and test the robustness and performance of tools and systems. It allows for the development and testing of security protocols, monitoring systems, and troubleshooting procedures without the need for sensitive or proprietary real data. By using synthetic log data, businesses and researchers can ensure that their applications and systems can handle real-world complexities and data volume.

## FUTURE SCOPE

In the field of artificial intelligence and machine learning, synthetic logs will play a crucial role in training and fine-tuning models, especially when real-world data is limited, sensitive, or unbalanced. Cybersecurity stands to benefit significantly, as synthetic log data enables the simulation of attack scenarios and the testing of security systems in a risk-free environment. In software development, it offers opportunities for robust testing of edge cases, scalability assessments, and continuous integration pipelines without the constraints of accessing real logs. Moreover, synthetic log data ensures compliance with stringent data privacy regulations like GDPR by enabling the analysis of representative data without exposing sensitive information. Emerging technologies such as digital twins, blockchain, and quantum computing will leverage synthetic logs to simulate realistic conditions and predict outcomes. Additionally, industries like education, healthcare, and IoT will use synthetic logs for training, simulation, and performance optimization. As tools and algorithms for generating synthetic log data evolve, its adoption will grow, addressing challenges like realism, scalability, and cost-efficiency while driving advancements across diverse sectors.

## REFERENCES

[1] "Distributed File System: A Review on Indexing and Searching Techniques" Authors: S. Sharma, A. Rana, S. Singh, S. Pandey Journal: 2019 *3rd International Conference on Computing Methodologies and Communication (ICCMC) Year: 2019*.

[2] Fei Bua, N.W.B.J.H.L. “Privacy by Design” implementation: Information system engineers’ perspective. *Int. J. Inf. Manag. 2020, 53, 102124.* [Google Scholar]

[3] Baldassarre, G.; Giudice, P.L.; Musarella, L.; Ursino, D. The MIoT paradigm: Main features and an “ad hoc” crawler. Future Gener. Comput. Syst. 2019, 92, 29–42. [Google Scholar] [CrossRef]

Power System Technology
ISSN:1000-3673

Received: 06-11-2024 Revised: 15-12-2024 Accepted: 05-01-2025

[4] Mukherjee, M.; Matam, R.; Shu, L.; Maglaras, L.; Ferrag, M.A.; Choudhury, N.; Kumar, V. Security and privacy in fog computing: Challenges. *IEEE Access 2017, 5, 19293–19304. [Google Scholar] [CrossRef]*

[5] Xie, J.; Qian, C.; Guo, D.; Wang, M.; Shi, S.; Chen, H. Efficient indexing mechanism for unstructured data sharing systems in edge computing. In Proceedings of the *IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 820–828. [Google Scholar]*

[6] Sunhare, P.; Chowdhary, R.R.; Chattopadhyay, M.K. Internet of things and data mining: An application oriented survey. J. King Saud Univ. Computer. Inf. Sci. 2020. [Google Scholar] [CrossRef]

[7] Busany, N.; van der Aa, H.; Senderovich, A.; Gal, A.; Weidlich, M. Interval-Based Queries over Lossy IoT Event Streams. ACM Trans. Data Sci. 2020, 1, 1–27. [Google Scholar] [CrossRef]

[8] "Integrated Indexing and Searching in Distributed File Systems: A Survey" Authors: S. Gautam, A. Rana Journal: 2018 *8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) Year: 2018.*

[9] W. Zhang, H. Tang, S. Byna, and Y. Chen. DART: Distributed Adaptive Radix Tree for Efficient Affix-Based Keyword Search on HPC Systems. In Proceedings of the *27th International Conference on Parallel Architectures and Compilation Techniques, PACT '18, 2018.*

[10] Zhou, Z., Wang, Y., & Liu, L. (2018). Log-based anomaly detection using synthetic log data. In 2018 *IEEE International Conference on Data Mining (ICDM '18), 1237-1242.*

[11] Abdullah, A. (2020). Synthetic log generation for large-scale security testing. *International Journal of Computer Science and Information Security, 18(5), 45-56.*

[12] Shetty, S., & Soni, P. (2021). A review of techniques for synthetic log data generation. *International Journal of Computer Applications, 174(10), 32-38.*

[13]Ravichandran, A., & Sundaram, N. (2022). Synthetic data for building realistic log analysis systems. *Journal of Software Engineering and Applications, 15(8), 112-118.*

[14]Sharma, P., & Agarwal, N. (2023). Using AI-driven synthetic log data generation for enhancing system monitoring. *Journal of Cloud Computing: Advances, Systems and Applications, 12(1), 71-84.*

[15]Moor, R., & Thomas, P. (2024). AI and synthetic data in anomaly detection for log-based systems. 2024 *IEEE International Conference on Machine Learning and Applications (ICMLA), 850-857.*