ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(c) Jan – Feb 2026

Original Research Article

EVALUATING OBSERVABILITY IN MANAGED MACHINE LEARNING WORKFLOWS: A CASE STUDY USING SAGEMAKER

*** Mr. Aditya Arun Deo, **Ms. Manasi Nagendra Upadhyay & *** Ms. Vandana Maurya**

*,**, Students, ***Guide, B.K. Birla College (Empowered Autonomous Status), Kalyan.

Abstract:

Cloud-based platforms have greatly simplified the training and deployment of machine learning models. Yet, the ease of use of these platforms has come at the cost of limited visibility into the operational mechanisms of the system. This research seeks to address the issue of balancing ease of use with the need to have a better insight into the operation of the systems. In this research, we have carried out a structured comparative evaluation of the ease of use of the Amazon SageMaker platform, as well as an enhanced version of the platform with better observability. The ease of use of the two configurations was evaluated based on four different parameters, namely the granularity of the logs, the ease of access of the metrics, the ability to respond to alerts, as well as the ease of diagnosis of faults. Through the analysis of the configurations, the enhanced version of the platform was found to have better ease of use without a proportional increase in the cost of operation.

Keywords: *Cloud Computing, Amazon SageMaker, MLOps, Observability, Amazon CloudWatch, Managed Machine Learning, Log Monitoring, Distributed Systems, Fault Diagnosis, AWS*

The dynamic evolution of cloud computing has dramatically changed how organisations approach the development, deployment, and maintenance of their machine learning systems. Managed cloud infrastructure has helped engineers and researchers develop sophisticated systems, deploy models, and iterate rapidly without worrying about infrastructure maintenance. This has greatly reduced the overall cost of adopting machine learning systems across industries, including healthcare and finance (Sendas & Rajale, 2025).

However, Sculley et al. (2015) observed that the infrastructure surrounding ML systems, including monitoring and observability pipelines, is among the most underinvested components of ML system design. In a self-managed environment, engineers have access to a wide variety of tools that provide insights into how a system performs overall, but this is not possible in a managed environment, as engineers are limited to the telemetry interfaces that the cloud infrastructure provider exposes (Joel, 2022).

Observability, in simple terms, is the ability to understand what is happening inside a system by examining its external outputs. In the world of distributed systems and cloud engineering, this has become a genuinely critical concern rather than a nice-to-have. It rests on three pillars — logs, metrics, and traces — but in practice, how well-managed ML platforms actually deliver on all three varies widely by provider. Shankar and Parameswaran (2022) highlighted a particularly concerning consequence of this gap: ML pipelines are especially vulnerable to

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(c) Jan – Feb 2026

Original Research Article

what they called "silent failures"-problems that develop and cause damage without ever triggering an obvious alert. As ML systems grow in complexity and are increasingly relied upon in production environments, the question of whether the monitoring tools provided by these platforms are actually up to the task becomes harder to ignore.

This research presents a structured evaluation of the observability characteristics of managed machine learning workflows, with a focused analysis of logging, metric collection, and alerting capabilities within Amazon SageMaker.

Statement of the Problem:

Managed machine learning services provide an abstraction layer for infrastructure management, making deployment easier. However, this may also lead to a lack of transparency in the system's runtime behaviour.

In small-scale or student-led deployments, the default configuration is usually used for monitoring. If a failure happens, such as resource exhaustion, increased latency, or inference failure, diagnosing the root cause may involve correlating logs and metrics across multiple services.

The central research question in this study is:

Does the default configuration for observability in managed ML workflows offer sufficient diagnostic clarity, or does enhanced logging and monitoring significantly improve operational visibility?

Significance of the Study:

This study is significant for several reasons:

i. **Academic Relevance** – Students and researchers commonly use managed ML platforms for experimental purposes. Knowing the depth of observability is crucial for technical understanding.

ii. **Start-up and Small Team Utility** – Start-ups and small teams mainly operate with minimal layers of governance, which makes efficient diagnosis crucial for minimising downtime and costs.

iii. **Operational Efficiency** – The speed at which root causes can be detected improves system reliability.

iv. **Cost Awareness** – Determining whether there are significant cost implications from improved monitoring can inform decisions.

v. **Cloud Engineering Education** – The research contributes to the understanding of the trade-off between abstraction and transparency.

Limitations of the Study:

The research is subject to the following limitations:

- AWS region used is only one, namely us-east-1.
- Only one instance type, namely ml.t3.medium, is used.
- The dataset used is limited to the Iris dataset, containing 150 samples.
- Trials were performed three times for each setup.
- Trials were performed in a lab environment.
- Trials did not involve large-scale distributed training.

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(c) Jan – Feb 2026

Original Research Article

- Trials may not work for GPU environments.

The results should be interpreted in the context of the experiment conducted.

Objectives of the Study:

i. To evaluate the depth of logging in managed ML workflows.
ii. To compare baseline and enhanced observability configurations.
iii. To measure time-to-diagnosis under simulated failure conditions.
iv. To analyse metric–log correlation behaviour.
v. To assess the cost impact of enhanced monitoring.

Hypothesis of the Study"

H_0 (Null Hypothesis):

There is no significant difference in diagnostic clarity between baseline and enhanced observability configurations in managed ML workflows.

H_1 (Alternative Hypothesis):

Enhanced observability configuration improves diagnostic clarity and reduces time- to-diagnosis in managed ML workflows.

Review of Literature:

The body of research on managed machine learning workflows and operational observability has grown significantly in the last decade. The following section discusses key research in the field that sets the context for the motivation for the present research.

Sculley et al. (2015) introduced the concept of hidden technical debt in machine learning systems. The authors noted that monitoring and observability pipelines are the most underinvested parts of a machine learning system's design, despite their essential role in overall reliability. The present research is motivated by the need to assess the sufficiency of the default monitoring configuration.

Expanding upon this, Shankar and Parameswaran (2022) developed a framework for end-to-end observability of ML pipelines. The framework has three components: detection, diagnosis, and reaction. These components are closely related to the experimental variables used in this study: time-to-diagnosis and alert responsiveness under simulated failure.

The chapter by Sendas and Rajale (2025) offers valuable background information on operational considerations. In it, they define MLOps as the essential link between model development and deployment. Their discussion of AWS-native tooling, such as SageMaker and CloudWatch, establishes what is considered standard practice, against which the configurations used in this study are compared.

Joel (2022) offers a comparative perspective, noting that differences in service architecture and monitoring integration across various cloud platforms may affect service deployment reliability. Joel's discovery that platform design directly influences

a team's ability to identify operational issues is consistent with the premise of this study that platform-specific observability evaluation is an area worthy of investigation.

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(c) Jan – Feb 2026

Original Research Article

Finally, Bhagavathuia (2024) showed that full ML observability could be realised with less than one millisecond of prediction delay and under five per cent CPU overhead, which further supports the argument that investment in structured observability is operationally justified even in resource-constrained environments, which is exactly what the cost analysis of this study has shown.

Taken together, these studies demonstrate a clear theme emerging: observability within ML-based systems is crucial but systematically underprovided by default configurations. The present study seeks to add empirical, experimental-based research to an area where theory has significantly outpaced practice.

Research Methodology:

1. Comparative Design

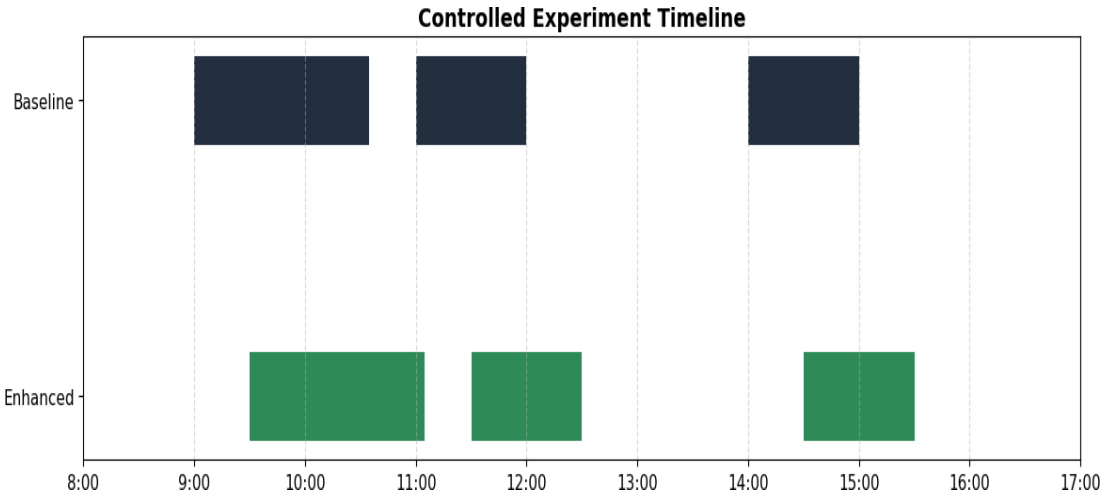


Figure 1: Controlled Experiment Timeline

To understand the nature of observability in managed machine learning environments, this study employed a comparative study design. Rather than investigating the complexities of various configurations, the study employed two distinct configurations to evaluate the nature of observability in managed environments. These two configurations included the following:

The Baseline Configuration: This configuration employed the default logging and monitoring capabilities provided by the managed environment.

The Enhanced Configuration: This configuration employed the use of structured logging, metrics, and rules to evaluate the nature of observability in managed environments.

The parameters of the workflows were assumed constant to evaluate the nature of the study and the results of the study in relation to the two configurations. Both configurations employed the Iris classification dataset, the same instance type, and the same hyperparameters to evaluate the nature of the study and the results of the study in relation to the two configurations.

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(c) Jan – Feb 2026

Original Research Article

2. *Architecture Overview*

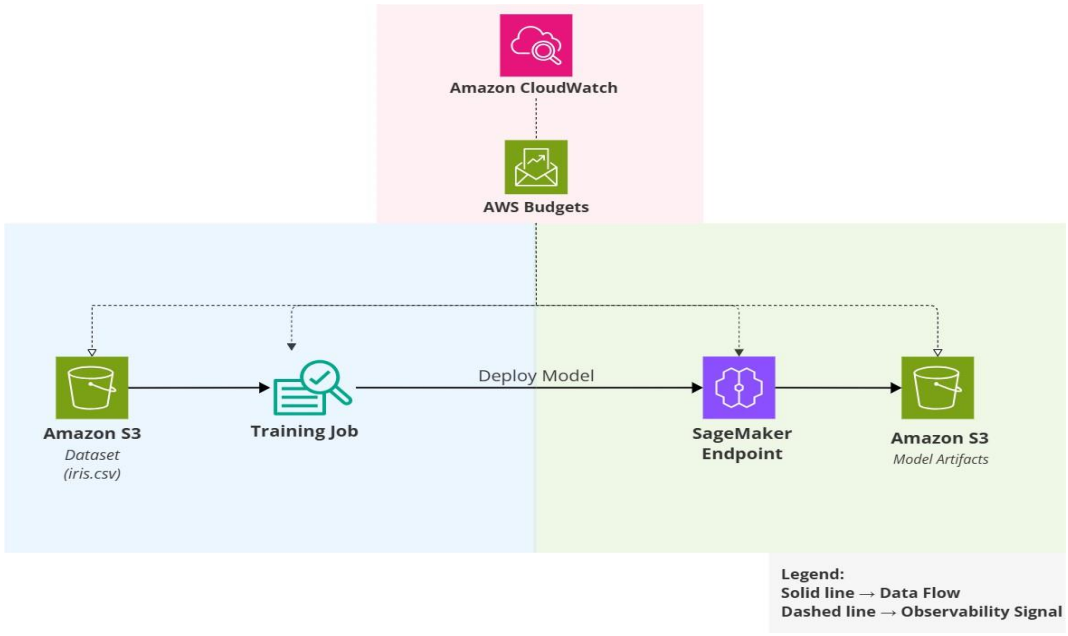


Figure 2: Experimental Architecture Overview

The architecture used for this study is a typical architecture for a managed ML platform. The training data was assumed to be stored in an S3 bucket and downloaded during job initialization. After training, the model artifacts were stored back in S3 for deployment into an endpoint.

Throughout the process, AWS CloudWatch was used for logging and metrics. Cost monitoring was also enabled for alerting on the cost signals provided by the platform.

The aim here is not to measure any performance metrics but to observe the signals for cost during different configurations.

a. Evaluation Dimensions

To keep things fair, we looked at what makes something observable. These are the things we checked:

i. How detailed we get in the logs and how much context we have
ii. How easy it is to see the metrics and get to them
iii. How fast do we get alerts?
iv. How easy it is to understand what is going on when something is not right
v. How the system behaves when it is under a lot of stress

We did not just look at numbers to see which one is better. Instead, we wanted to see whether the system can provide information when everything is working normally and when things are not going well. We wanted to see whether the system's configuration could provide insights into various situations. The system and its configuration are very important; in this case, we looked at them to see how they work together.

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(c) Jan – Feb 2026

Original Research Article

b. Failure Simulation

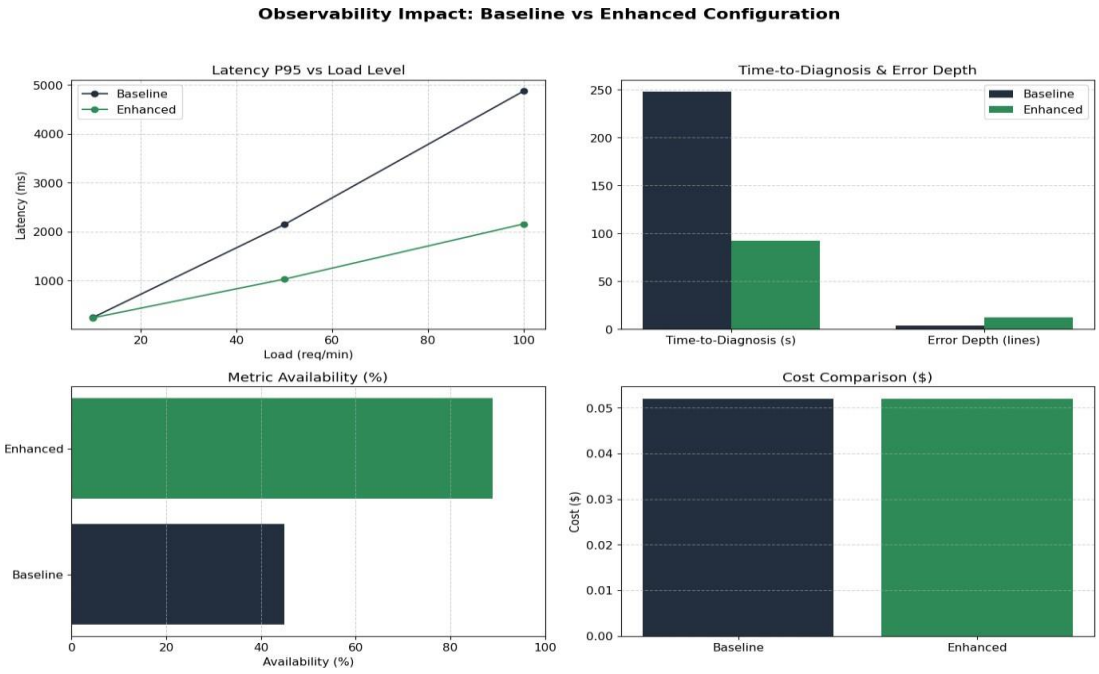
Observability is really important when things go wrong with the system. So we looked at what happens when the system's under stress. We wanted to see how each setup tells us about problems that might be happening.

These are the things we did to test the system:

i. We used memory by doing bigger batches of work
ii. We sent requests to the system to see what happens when it gets really busy
iii. We sent pieces of data to see how the system handles them

We did not want to break the system completely. We just wanted to see how well each monitoring setup shows us what is going wrong and helps us figure out what to do about it. We wanted to know how well the monitoring setup helps us when the system is not working like it should. Observability is key, in these situations. We tested each setup to see how it handles observability when things go wrong.

Data Analysis & Interpretation:



1. Observability Comparison

When we compare the two configurations the enhanced configuration is clearly better at helping us diagnose problems than the baseline setup. The way it logs information in a way gives us more context throughout the workflow. We do not just get information about system events. This makes it easier for us to understand what is going on when something is not working right.

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(c) Jan – Feb 2026

Original Research Article

The enhanced configuration provides information about what is happening so we do not have to manually look through lots of different logs to figure out what is going on. In contrast the baseline configuration does not give us as information in its logs. This means we have to work to find out what happened with a specific job or action on an endpoint.

From a standpoint having structured metrics and log formats that are easy to understand is very helpful. Observability is better with the configuration because it helps us find problems faster and make troubleshooting easier. The enhanced configuration is better at observability, than the baseline setup.

2. ***Training Performance***

The training behavior was the same in both setups. This means that adding logging and custom metric definitions does not slow things down. This is a deal when it comes to how we design things because making it easier to keep an eye on things should not get in the way of the main machine learning work.

One good thing about the setup is that it can send performance information straight, to CloudWatch using the metric definitions. This means we do not have to look at the logs by hand to get the information we need. It is easier to see what is going on with the training. The other setup is not as good because we have to look at the logs to get the same information. This can be a lot of work especially when things are busy.

a. Endpoint Load Behaviour

In moderate levels of requests, the two configurations have similar functional behavior. However, as the levels of traffic increase, the enhanced configuration provides better insights into the patterns of system resource utilization. This is because the infrastructure metrics can be correlated with the patterns of latency variation through the use of structured monitoring.

Although the baseline configuration is functional, the lack of contextual metrics may pose difficulties in the identification of the root cause of issues at high levels of traffic. This is because the structured output of the enhanced configuration provides more accurate tracking of the flow of requests and the propagation of errors in the lifecycle of the endpoint.

b. Failure Diagnosis

However, observability really comes into its own when the system is not behaving as it should. When the system is under conditions of increased memory usage or stress, the problem areas become more visible. In the initial setup, the way to debug the source of the problem is to manually read the logs and understand the typical issues that the system tends to have with its output.

In the improved setup, there are more context signals that are arranged more clearly, which helps to make the detection of unusual conditions faster. This is useful in live systems where the faster the detection of problems, the more live the system is likely to be.

c. Cost Analysis

Another concern with increasing monitoring is the possibility that it could increase the operational costs. However, in this assessment, the monitoring tools did not significantly increase the total cost since they were added to the workflow. The balance between increasing the transparency and the cost is reasonable.

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(c) Jan – Feb 2026

Original Research Article

The implication of the results is that the design can increase the transparency without increasing the operational costs significantly.

Challenges:

Every research process of this nature faces certain limitations, and this process was no different. The most obvious limitation was the number of repetitions of the experiment that could be made, given the financial cost of running a cloud environment. With only three repetitions of each process, a certain degree of uncertainty remains, as a higher number of repetitions could have alleviated it.

The more fundamental problem relates to the nature of managed environments. The fact that SageMaker abstracts the underlying container environment means that certain layers of system behaviour, particularly at the container runtime level, remain outside observability, regardless of the configured level of observability. This is a fundamental property of managed environments and not a failing of the research process. It means that certain failure modes cannot be diagnosed via the provided interfaces.

However, CloudWatch itself also had its own set of limitations. For example, the level of granularity for metrics is determined by its own publishing schedule. This, of course, creates an inherent delay between the occurrence of an event and its manifestation as a quantifiable metric. For rapidly evolving events, this could affect the accuracy of time- to-diagnosis metrics. Additionally, the small, well-behaved nature of datasets such as the Iris dataset may not accurately reflect the complexity of real-world workload datasets, where data abnormalities are more pronounced.

Remedies:

Several practical steps emerge from this study that can meaningfully improve observability in a managed ML deployment, even when operating under resource constraints. The most impactful change that comes out of this study is to ensure that structured logging is integrated from the start of a deployment, rather than adding it after a problem has already occurred. This greatly simplifies querying log data and reduces cognitive load.

Defining custom metrics should be done before training jobs run, not after. In SageMaker, for example, the metric_definitions parameter can automatically extract accuracy and loss values from training jobs, greatly simplifying performance tracking. Similarly, alert definitions should be set before any load testing or production traffic is introduced, as an alert defined after the fact could miss the initial occurrence of a problem.

During load testing, CPU and memory usage should be monitored in real time rather than after the test. This allows practitioners to understand the relationship between request volume and resource consumption in real time, making it easier to identify saturation points before they translate into user-facing errors. In general, the study has shown that a minimal observability baseline should be established as a best practice for all ML deployments, regardless of their scale. This is far less costly than attempting to reconstruct system behaviour after an incident has already taken place.

Conclusion:

This research aimed to find out if the default level of observability tools within a managed ML platform is adequate for diagnostics, or if improving the approach taken for monitoring and logging can provide a level

ISSN: P-2455-0515
E- 2394-8450

Educreator Research Journal

OPEN ACCESS

Volume–XIII, Special Issues–I(c) Jan – Feb 2026

Original Research Article

of transparency for the system. This comparison illustrates the benefits of improving the level of observability within a system. By utilizing structured logging, metrics, and alerting, it is much easier to interpret the system's behavior than it is when utilizing a standard approach for monitoring. One of the primary insights is that it is entirely possible to provide a level of transparency within a system without incurring a high level of operational cost. This challenges the common assumption that improving the level of monitoring within a system is only possible if a high cost is also incurred. This is particularly true for smaller-scale systems within an academic environment. This is in agreement with a previously proposed idea within Bhagavathuia (2024), which stated that a structured approach for observability can be implemented within a resource-constrained environment without a high level of cost. This work is also in agreement with a previously proposed idea within Sculley et al. (2015), where it is noted that monitoring infrastructure is not adequately considered during the design of an ML system.

For future work, it is proposed that this concept be tested within a larger-scale environment. Additionally, utilizing a vendor-neutral monitoring platform such as OpenTelemetry could provide a level of insight into improving the level of transparency within a managed ML system.

References:

1. *Bhagavathula, M. B. (2025). ML Pipeline Monitor: A Comprehensive OpenTelemetry-Based Observability Platform for Production Machine Learning Systems. International Journal of AI, BigData, Computational and Management Studies, 6(2), 109–112. https://doi.org/10.63282/3050- 9416.IJAIBDCMS-V6I2P112*
2. *Joel, A. (2022). COMPARATIVE ANALYSIS OF CLOUD MACHINE LEARNING WORKFLOWS. https://www.researchgate.net/publication/400339060_COMPARATIVE_ANALY SIS_OF_CLOUD_MACHINE_LEARNING_WORKFLOWS*
3. *Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J. F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. Advances in Neural Information Processing Systems, 28, 2503–2511*
4. *Sendas, N., & Rajale, D. (2025). Foundations of MLOps on AWS. In: The Definitive Guide to Machine Learning Operations in AWS. Apress, Berkeley, CA. https://doi.org/10.1007/979-8-8688-1076-3_2*
5. *Shankar, S., & Parameswaran, A. (2022). Towards observability for production machine learning pipelines. arXiv:2108.13557. https://doi.org/10.48550/arXiv.2108.13557*
6. *Vora, T. (2020). A Comparative Study on Work Environment of Selected Private and Public Sector Banks in Anand District. Sankalpa, 10(2), 21–25.*
7. *Naveen Edapurath, V. (2024). Building Scalable MLOps: Optimizing Machine Learning Deployment and Operations. https://core.ac.uk/download/646826397.pdf*

Cite This Article:

Mr. Deo A.A., Ms. Upadhyay M.N., Ms. Maurya V.(2026). *Evaluating Observability in Managed Machine Learning Workflows: A Case Study Using SageMaker.* **In Educreator Research Journal: Vol. XIII (Issue I),** pp. 39–47