

Proactive Forensics for Online Exam Using Log Management Approach

A THESIS SUBMITTED TO
THE SCHOOL OF COMPUTING

BY

Priyagung Elza Yogitama

2302221005



IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF FORENSIC SCIENCE
IN
THE SCHOOL OF COMPUTING

TELKOM UNIVERSITY
2025

APPROVAL PAGE

Approval of the School of Computing of Telkom University

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master Forensic Science.

Date Jul 03 , 2025 (*the date can be set manually)

(Dr. Farah Afianti)

Head of Master Forensic Science

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Forensic Science.

Date Jul 03 , 2025

(Supervisor's name)

Supervisor

(Co-Supervisor's name)

Co-Supervisor

Examining Committee Members.

Date Jul 03 , 2025

(Jury's name) (Chairperson of the jury)

: _____

(Jury's name) (jury's member)

: _____

(Jury's name) (jury's member)

: _____

SELF DECLARATION AGAINST PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Month/Date/Year Jul 03 , 2025

Name, last name: Priyagung Elza Yogitama

Signature: _____

Month/Date/Year Jul 03 , 2025

Name, last name of the Supervisor: NIKEN DWI WAHYU CAHYANI, S.T., M.Kom, Ph.D.

Signature: _____

Month/Date/Year Jul 03 , 2025

Name, last name of the Co-Supervisor: Assoc. Prof. Dr. VERA SURYANI, S.T., M.T.

Signature: _____

ABSTRACT

ABSTRAK

Kata kunci:

DEDICATION

This thesis is compiled with the support of my family. I praise GOD and thanks to all who have helped, directed, and supported this work. I dedicate the work to my beloved parents ALICE and BOB.

Finally this thesis is dedicated to all informatics students in the world. I hope that this research may provide valuable contribution in Computer Science.

ACKNOWLEDGMENTS

This thesis is compiled with the effort, help, and support from both students and lecturers. I would like to express my deepest gratitude and thanks to:

1.
2.

PREFACE

In thesis writing, the most difficult part to write is Chapter 1 (Introduction/The Problem). As they say, the most difficult part of any endeavor is the starting point. This is because the first chapter is where you conceptualize your entire research. The whole research/thesis can be reflected in Chapter 1 including expected results or outcomes. For your guidelines, please read the following sample format of Chapter 1.

Bandung,

Priyagung Elza Yogitama

CONTENTS

APPROVAL	ii
SELF DECLARATION AGAINST PLAGIARISM	iii
ABSTRACT	iv
ABSTRAK	v
DEDICATION	vi
ACKNOWLEDGMENTS	vii
PREFACE	viii
CONTENTS	ix
LIST OF TABLES	xi
LIST OF FIGURES	xii
LIST OF TERMS	xiii
1 INTRODUCTION	1
1.1 Rationale	2
1.2 Theoretical Framework	2
1.3 Conceptual Framework/Paradigm	2
1.4 Statement of the Problem	2
1.5 Objective and Hypotheses	3
1.6 Contribution	4
1.7 Assumption	4
2 REVIEW OF LITERATURE AND STUDIES	5
2.1 Related Literatures	5
2.2 Related Studies	7
2.2.1 Digital Forensics	7
2.2.2 Log Management	9
2.2.3 Learning Management System	9
3 Research Methodology	11
3.1 Research Design	11
3.1.1 System Implementation	11

3.1.2	Log Management Implementation	20
3.1.3	Log Management Testing Phase	21
4	PRESENTATION, ANALYSIS AND INTERPRETATION OF DATA	23
4.1	Phase 1	24
4.1.1	Define Problem Statement	24
4.1.2	Research Objectives	26
4.1.3	Research Method	26
4.2	Phase 2	27
4.3	Phase 3	28
4.4	Phase 4	29
4.4.1	Log Identification	30
4.4.2	Proactive Collection	38
4.4.3	Log Transmission	39
4.4.4	Log Storage	40
4.4.5	Analysis of the Data	41
4.4.6	Log Monitoring	47
4.4.7	Notification	49
4.4.8	Log Preservation	50
4.4.9	Reporting	53
4.5	Phase 5	53
4.5.1	Verification	54
4.6	Phase 6	55
4.6.1	Validation	55
4.7	Result of framework log management	55
4.8	Summary of Findings	58
5	Conclusion and Recommendations	61
5.1	Conclusion	61
5.2	Recommendations	61
	BIBLIOGRAPHY	63
	Appendices	65
A	Hasil Wawancara dengan Staf IT	67

LIST OF TABLES

2.1	Overview of Research to Online Examination System and Forensic Technique	6
3.1	Stages of Framework Development	14
4.1	Summary of Related Works and Relevance to This Research	28
4.2	Table Log Source, Description and Example Log Data	31
4.3	Course Log Records from User Attempts Quiz	36
4.4	Course Log Records from Moodle Course	37
4.5	Example of column names and corresponding data in the online exam anomaly de- tection system	42
4.6	Features Used for Analysis Data	43
4.7	Classification Report with Precision, Recall, and F1-score	46
4.8	Example of standard_log table structure	51
4.9	Quiz Attempt Log Data Example	52
4.10	Backup File Information	52
4.11	Verifying testing framework adopted from NIST 800-92	54
4.12	Comparison of Proactive and Reactive Forensic Based on Log Management Aspects	58
4.13	Log Management Processes	59

LIST OF FIGURES

2.1	Phase log management from NIST 800-92	7
2.2	Forensic Readiness, Proactive Forensics and Reactive Forensic	8
2.3	Functional process for proactive and reactive digital forensics investigation system [4]	8
3.1	Reference architecture of the university's online examination system	11
3.2	Architecture of Azure VM Scale Set used in the implementation	12
3.3	Proactive Forensic Framework Development Phase Adel et al. [2]	13
3.4	Proactive Forensic Framework Development Phase 1	15
3.5	Proactive Forensic Framework Development Phase 2	16
3.6	Proactive Forensic Framework Development Phase 3	17
3.7	Proactive Forensic Framework Development Phase 4	18
3.8	Proactive Forensic Framework Development Phase 5	19
3.9	Proactive Forensic Framework Development Phase 6	20
4.1	Log Management Structure Menu	23
4.2	Digital Forensic Evidence Loss [17]	25
4.3	Log Source	30
4.4	ERD from Several Column Tables	35
4.5	Backup from mysql dump	36
4.6	Scheduler for proactive collection	38
4.7	Exporter script	39
4.8	Machine learning workflow	40
4.9	Log data backup	41
4.10	Flowchart ML Training	44
4.11	Flowchart Log Analysis	45
4.12	Log Attempt Step	48
4.13	Finding user	48
4.14	Notification	49
4.15	Log Preservation Directory	50
4.16	Detail Log Preservation	50
4.17	Reporting User	53
4.18	Reporting User Download	53
4.19	Phase log management from NIST 800-92 2006 Kent and Souppaya [13]	55
4.20	Proactive Forensics Alharbi et al. [4]	56
4.21	Proposed framework adapted from NIST 800-92	57

LIST OF TERMS

Terms	Definition
Classes	Number of individual in biometrics data
Sample	Number of images can be used to represent population in a class.
...	...

CHAPTER 1

INTRODUCTION

To provide learning resources, educational institutions use online learning platforms. Students can engage in learning activities and communicate with one another using these technologies. Distant web locations. Security risks are raised by this behavior. Additionally, worries about the validity of the online test procedure have grown, which is one of activities that are essential for online learning [26].

Technological advancements are rendering traditional digital forensics techniques and tools potentially obsolete. Proactive approaches, emphasizing remote investigation capabilities, are emerging as a new paradigm to address this challenge. Machaka and Balan [19]. Digital forensics underpins effective cybersecurity incident response. Its methods enable reconstruction of attack sequences, providing a clear understanding of malicious actions. This forensic analysis empowers responders to contain threats and implement mitigation strategies, while also potentially yielding legally valuable evidenceJohansen [11].

The continuous advancement of information technology has witnessed a surge in the adoption of electronic examination (e-exam) systems by researchers and organizations. Concurring with prior research Al-Fayoumi and Aboud [3], institutions employing e-exam systems must prioritize robust security measures. These safeguards are paramount for ensuring the confidentiality, integrity, and availability of examination information, ultimately protecting the institution's reputation. One method for gathering logs indicative of online exam cheating is by utilizing log management.

In "Guide to Computer Security Log Management," NIST Special Publication 800-92, a thorough framework for creating and sustaining efficient log management procedures is presented. Given that it describes methods for gathering, examining, and maintaining digital evidence, this approach is especially pertinent to tackling the problems associated with online exam cheating.

Reactive forensics, which investigates incidents post-occurrence, poses significant risks in addressing online exam threats. Delayed detection can lead to greater damage to exam integrity and institutional reputation, along with the potential loss of critical evidence, such as deleted log data. This approach is not only resource-heavy, demanding extensive time and effort, but also lacks preventive capabilities, allowing misconduct to persist. As online exam platforms grow, reactive methods may struggle to manage the increasing volume of data and incidents effectively.

In order to solve these problems, proactive forensics uses anomaly detection, log management, and real-time monitoring to spot dangers as they materialize. The security and integrity of online tests are guaranteed by frameworks such as NIST 800-92, which allow institutions to quickly identify and prevent evidence data loss.

The purpose of this research is to explore the application of proactive forensic techniques, guided by NIST 800-92, in addressing challenges such as cheating and system abuse in online exam systems. By integrating log management with real-time anomaly detection, this study aims to enhance the

security, reliability, and integrity of e-exams, offering practical solutions for modern educational institutions.

1.1 Rationale

The growing usage of Learning Management Systems (LMS) in training and education has made the problem of cheating on online tests more pertinent. Participants benefit from increased flexibility and accessibility while taking exams online, but maintaining exam integrity also becomes more difficult. Consequently, it is crucial in this situation to apply proactive forensics using a log management strategy. Given the complexity of today's fraud problems, a reactive approach for cheating detection where analysis is done after an incident has happened is no longer enough. Because proactive forensics can identify suspicious activity in real time, it provides a more sophisticated solution that enables institutions to take action before cheating happens.

1.2 Theoretical Framework

Log management is the primary instrument used in this study's proactive forensic method to gather and examine log data that can indicate questionable activity in an online English proficiency test. By preserving the integrity of the data and the online test procedure, this method seeks to preventively identify possible fraud.

This study, which is grounded in log management theory, uses user activity logs including access time, duration, and navigation patterns to identify unusual activities. Effective preventative measures and risk mitigation are designed using information security and digital forensic theories as a guide to guarantee the process's security and validity. The goal of integrating these theories is to preserve the security and integrity of online tests while actively preventing cheating.

1.3 Conceptual Framework/Paradigm

Identify and discuss the variables related to the problem, and present a schematic diagram of the paradigm of the research and discuss the relationship of the elements/variables therein.

1.4 Statement of the Problem

The issue of cheating in online exams has become increasingly relevant with the increased use of Learning Management Systems (LMS) in education and training. Ranger et al. [23] mention about several indicator of cheating .The use of online exams provides greater flexibility and access to participants, but also creates new challenges in ensuring exam integrity.

1. Inefficient Log Management Existing log handling methods lack automation and centralization, leading to delays in collection, storage, and analysis, which undermines forensic readi-

ness.

2. Risk of Evidence Loss Without a robust system for log preservation, critical evidence may be lost or become unreliable, compromising forensic investigations in online examination platforms.
3. Lack of Alerts and Reporting The absence of integrated alert systems and reporting mechanisms limits timely detection of anomalies and documentation of potential irregularities for further analysis.

1.5 Objective and Hypotheses

Objective

The actual purpose of the proposed method for log management in proactive forensic systems would be to create a reliable and orderly structure for the acquisition, storing and analysis of log data. The method is likely to increase forensic preparedness by making the log data collection process automated and centralizing it towards a secured place, which will also ensure the integrity and availability of digital evidence and risk of losing data. Using the NIST 800-92 framework for security log management equips organizations with the tools and practices to handle logs effectively. It enhances forensic readiness

- Designing and deploying a log management system that centralizes log handling to enable efficient management, secure storage, and preparedness for forensic analysis within an online examination platform.
- Developing a system that automates and proactively collects, stores, and analyzes log data to enhance forensic readiness and ensure the availability of critical evidence for investigations.
- Incorporating a reporting mechanism to document potential irregularities and support further forensic analysis based on the automated log evaluation process.

Hypothesis

1. The efficiency and accuracy of log management will be increased by automating and centralizing the collecting of logs from different parts of an online examination platform, such as server logs, user activity logs, and LMS like Moodle. This will allow for better monitoring and the identification of suspicious activity.
2. Automating the processes of collecting, storing, and analyzing log data enhances forensic readiness by guaranteeing the reliable availability of crucial evidence for efficient and timely investigations.
3. The impact of security issues is reduced by integrating real-time alert systems, such as Telegram dashboards or notifications, which improve administrators' or proctors' capacity to identify, respond quickly and effectively to anomalies found.

4. A structured preliminary report for online tests that includes user activity patterns and in-depth log analysis can help identify anomalies and offer practical advice for maintaining the integrity of the test. Forensic investigations and more precise tracking of cheating will be made possible by effective data log management.

1.6 Contribution

Proactive forensic techniques, guided by the NIST 800-92 log management framework, provide a structured and efficient approach to securing and analyzing logs. On purpose evidence data loss for forensically sound

1.7 Assumption

1. Accurate Documentation of Participant Activities: The log system reliably documents all participant activities during the EPT exam.
2. Authenticity of Preserved Log Data: The log data is trustworthy for examining examinee behavior because it is gathered from trusted sources and hasn't been altered.
3. Logs Show Real User Behavior: The recorded behavior patterns show the examinees' real activities because the generated log data accurately depicts their actions during the test.

These presumptions serve as the foundation for proactive forensics research and application in identifying possible irregularities or cheating during online EPT tests.

CHAPTER 2

REVIEW OF LITERATURE AND STUDIES

2.1 Related Literatures

In previous studies, such as Sylla et al. [32], the preventive aspect of reducing cheating in online exams conducted through Learning Management Systems (LMS) was discussed, including methods like using Secure Exam Browser (SEB) and randomizing questions. Meanwhile, the paper by Ranger et al. [23] focuses on detecting indicators of cheating in online exams.

A study of previous studies was done in order to better understand how forensic techniques are applied. The chosen papers examine a range of forensic techniques, such as proactive and reactive methods.

Table 2.1: Overview of Research to Online Examination System and Forensic Technique

References	Year	Study Scope	Type of Forensics	Framework Used	Research Gap
Venter and Kigwana [33]	2018	DFR on Online Examination	Proactive Forensic	ISO 27043:2015	Using NIST 800-92
Kadoic and Oreski [12]	2018	Moodle usage and academic success on student behaviour analysis logs-based	Not forensic purpose	None	Using logs-based for forensic activity
Rivera-Ortiz and Pasquale [24]	2019	Forensic-ready logging systems	Proactive Forensics	OWASP	Using framework logging by NIST 800-92
Febriana et al. [8]	2023	Integrating Forensic Techniques into Incident Response in private cloud	Reactive	NIST 800-86 and ISO 27043:2015	Difference environment on scope academic and using logging by nist 800-92
Lakhno et al. [16]	2023	Information System Security	Proactive Forensic	NIST 800-92	Difference environment by academic purpose online exam
Kern et al. [14]	2024	Logging maturity	Proactive and Reactive Forensic	NIST 800-53	Using nist 800-92
Abd Hamid et al. [1]	2024	LMS with forensic logging	Proactive Forensics	Using framework by owasp describe 5W+1H	Using framework by NIST 800-92 for logging management
Lintang et al. [18]	2024	LMS activity during online tests	Proactive Forensics	None	using framework by nist 800-92 for proactive forensic purpose

The table 2.1 shows that different methodological techniques are employed to enable proactive forensics. To find research gaps that can be the basis for creating a proactive log management framework based on forensics, a comparison analysis has also been carried out. The main goal is to

combine log management techniques that improve preventive evidence loss with suggestions from the NIST 800-92 framework.

As an aspect I aim to develop from previous research, the lack of forensic readiness preparedness has been identified. Therefore, as shown in the image below, this represents the framework I have developed.



Figure 2.1: Phase log management from NIST 800-92

- **Log Generation:** The process where logs are created by operating systems, applications, and network devices to record events or activities.
- **Log Transmission:** The step where generated logs are transmitted to centralized storage or log management systems, either in periodically.
- **Log Storage:** Logs are stored in a secure and organized manner, ensuring availability, confidentiality, and integrity for future analysis.
- **Log Analysis:** Stored logs are analyzed to identify patterns, detect anomalies, investigate incidents, or support forensic investigations.

2.2 Related Studies

This section explores related studies that underpin the proposed method's design.

2.2.1 Digital Forensics

Digital forensics, also known as computer forensics, encompasses a wide spectrum of forensic investigations that extend beyond traditional computer devices. It incorporates expertise from various domains, including network forensics, database forensics, mobile device forensics, cloud forensics, memory forensics, and data or disk forensics.[21].

The digital forensic process has several general stages, such as, identifying, preservation, analysis and presentation. these phases were proposed by mckemmish in 1999.

Proactive Forensics: the ability to proactively collect, trigger an event, and preserve and analyze evidence to identify an incident as it occurs. In addition, an automated preliminary report is generated for a later investigation by the reactive component. Proactive evidence pertaining to a particular occurrence or incident as it happens will be acquired in this component [4]

Reactive Forensics: the conventional method of looking into a digital crime after it has happened, often known as the post-incident method [4]. Identification, preservation, collection, analysis, and production of the final report are all part of this process.

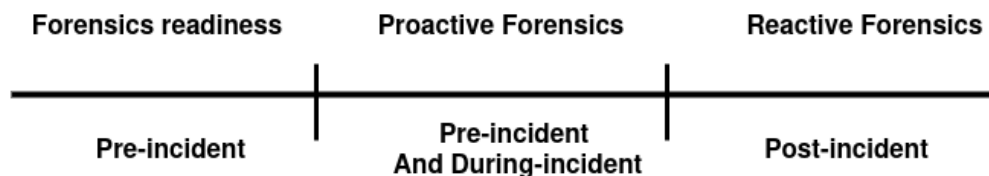


Figure 2.2: Forensic Readiness, Proactive Forensics and Reactive Forensic

Fig 2.2 above there is several section from forensic readiness, proactive forensics and reactive. The three methods mentioned above have a distinction in their application when dealing with cybercrime.

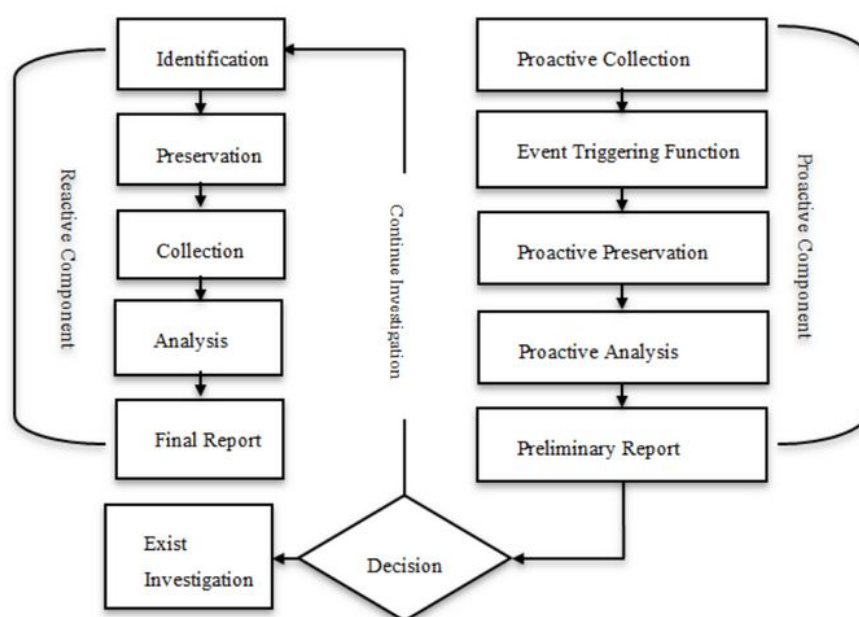


Figure 2.3: Functional process for proactive and reactive digital forensics investigation system [4]

In **Fig 2.3** Explain flow process the differences reactive and proactive forensics. Proactive collect the logs before the incident occurred.

- **Proactive Collection** : Automated live collection of a pre-defined data in the order of volatility and priority, and related to a specific requirement of an organization.
- **Even Triggering Function** : Suspicious event that can be triggered from the collected data.
- **Proactive Preservation** : Automated preservation of the evidence related to the suspicious event, via hashing.
- **Proactive Analysis** : automated live analysis of the evidence, which might use forensics techniques such as data mining to support and construct the initial hypothesis of the incident.

- **Preliminary Report:** Automated report for the proactive component.

Reactive methods struggle to identify and prove attacks efficiently. Proactive methods solve this by enabling real-time investigations and organizing data to save time and space. This makes proactive approaches faster and more effective for forensic investigations[27].

2.2.2 Log Management

Effective log management practices are critical for maintaining computer security records with the necessary detail and for an appropriate retention period, as outlined in NIST Special Publication 800-92 (SP 800-92). Additionally, log analysis proves beneficial for auditing, forensic analysis, internal investigations, establishing baselines, and identifying operational trends and long-term problems. [13]

The log management process encompasses a series of sequential steps, as outlined below:

1. **Log Generation:** The process where systems or devices generate logs. Logs can be created by various sources such as firewalls, servers, applications, etc. These logs record events, transactions, or activities that provide valuable information for security monitoring and auditing.
2. **Log Transmission:** The process of sending logs from the source (e.g., firewall, server, application) to a central storage or analysis system. This can be done via various methods, such as syslog, agent-based collection, or secure transmission protocols to ensure the integrity and confidentiality of the data.
3. **Log Storage:** The phase where logs are stored either in raw or structured formats. This could include storage in flat files, databases, or Security Information and Event Management (SIEM) systems. Proper storage ensures that logs are easily accessible for future analysis, audits, and incident investigations while maintaining their integrity.
4. **Log Analysis:** The phase of examining logs to detect anomalies, patterns, or security incidents. This involves using automated tools or manual review to identify suspicious activities, potential threats, or system malfunctions. Log analysis helps in proactive threat detection and reactive forensics during security incidents.

2.2.3 Learning Management System

Learning Management Systems (LMS) create virtual classrooms that enhance learning for both instructors and students. These online environments provide a framework for fostering an inclusive learning experience that supports academic progress. LMS tools encourage collaboration among students through online groups, discussions, and communication features, extending the benefits to all users [6].

Numerous studies have been conducted to evaluate and develop LMS. Smith and Brown [29] conducted a comparative study between Moodle and Google Classroom in higher education institutions. The study found that Moodle offers more features supporting structured learning processes, while Google Classroom is easier to use for both instructors and students.

Another research by Wijaya *et al.* [35] investigated the use of mobile-based LMS in distance learning. Their study found that the use of mobile-based LMS can increase student participation, especially during online learning in the COVID-19 pandemic.

Furthermore, Putra and Sari [22] explored the integration of artificial intelligence technology into LMS. The results indicated that AI-based recommendation systems in LMS can help improve students' motivation and engagement in the learning process.

However, most existing studies are still limited to aspects of usability and features, while issues related to scalability and data security in LMS for large-scale institutions have not been widely discussed [15].

CHAPTER 3

Research Methodology

3.1 Research Design

This research adopts a system development and experimental approach to implement and validate a proactive forensic framework for online examination systems. The proposed framework adheres to the structured log management principles outlined in the NIST SP 800-92 standard [13], aiming to establish a centralized, secure, and automated environment for collecting, analyzing, and preserving digital evidence.

The system is integrated with a Moodle-based Learning Management System (LMS), simulating an English Proficiency Test (EPT). Log data from quiz attempts and participant activities are systematically collected and analyzed using machine learning-based anomaly detection. Key features of the system include real-time monitoring, a web-based dashboard, and alert mechanisms to enhance forensic readiness.

3.1.1 System Implementation

The proactive forensic system includes a log acquisition stage using automated scripts to extract daily activity data. This data is stored in a designated log repository and analyzed using machine learning to detect anomalies. A dashboard web application is developed to visualize log activities and notify stakeholders of suspicious events via Telegram bot integration.

The system architecture is modeled after the existing infrastructure of a university's online examination environment, using Moodle LMS and Azure-based virtual machines. Figure 3.1 illustrates the reference architecture, while Figure 3.2 depicts the use of Azure Virtual Machine Scale Sets (VMSS) to ensure scalability.

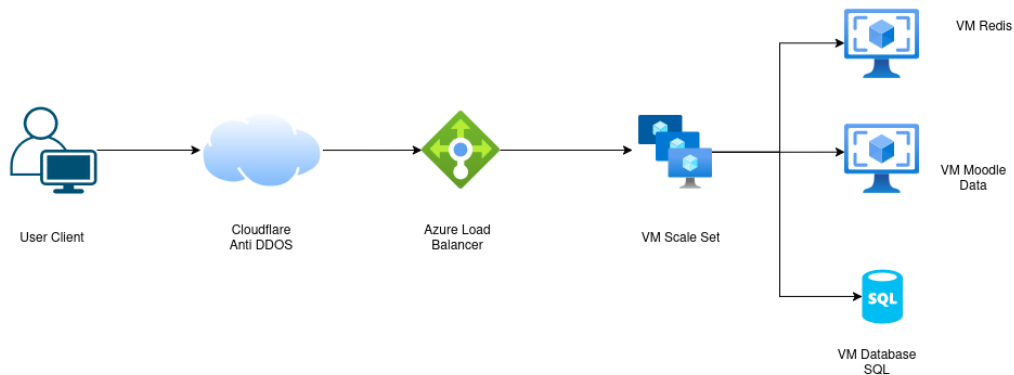


Figure 3.1: Reference architecture of the university's online examination system

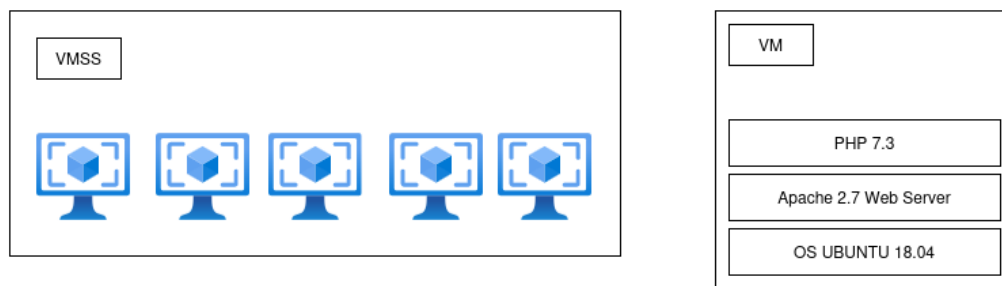


Figure 3.2: Architecture of Azure VM Scale Set used in the implementation

The system infrastructure utilizes a Virtual Machine Scale Set (VMSS) to dynamically manage computing resources during online examinations. VMSS enables automatic provisioning (*auto-create*) of virtual machines when an exam session begins, and de-provisioning (*auto-destroy*) once the session has ended.

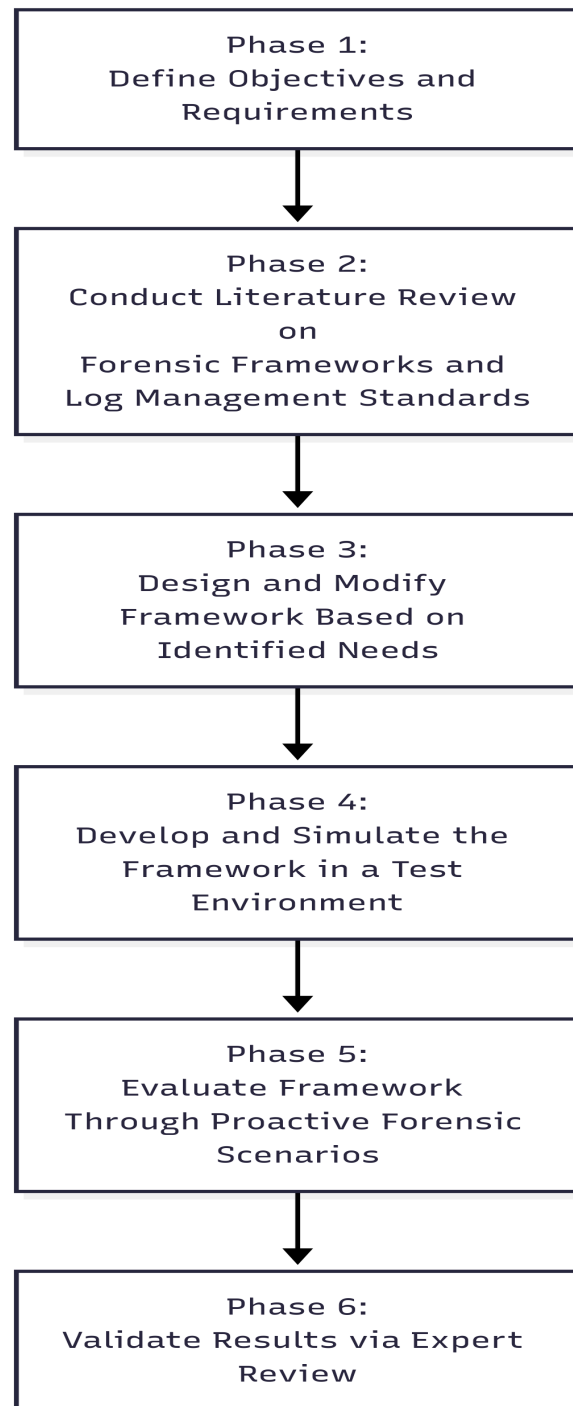


Figure 3.3: Proactive Forensic Framework Development Phase Adel et al. [2]

Figure 3.3 presents the proposed framework development flow, adapted from Adel et al. [2]. The framework consists of six phases: defining objectives, literature review, framework design, simulation, testing, and expert validation.

The last two phases testing and validation were added to evaluate the framework in real exam

scenarios and to gather feedback from domain experts in digital forensics and educational systems. This ensures both technical reliability and practical relevance in online examination contexts.

Table 3.1: Stages of Framework Development

No.	Stage	Description
1	Define Objectives and Requirements	Identify the primary goals and specific requirements for developing the forensic log management framework.
2	Conduct Literature Review on Forensic Frameworks and Log Management Standards	Study existing research, models, and relevant standards (e.g., NIST SP 800-92) to gain insight and identify gaps.
3	Design and Modify Framework Based on Identified Needs	Create or adapt framework components to match the defined objectives and contextual system needs.
4	Develop and Simulate the Framework in a Test Environment	Implement the proposed framework in a controlled testbed to observe log flow and system behavior.
5	Evaluate Framework Through Proactive Forensic Scenarios	Test the framework's effectiveness by applying it to simulated forensic incidents or attack scenarios.
6	Validate Results via Expert Review	Review and validate the framework through expert evaluation or feedback for reliability and improvement.

Based on the framework development approach proposed by Adel et al. [2], this research follows a structured process that begins with identifying existing forensic frameworks and relevant standards. The framework is then designed to address specific needs in digital forensics for online examination environments. To ensure that the framework is practical and effective, it is evaluated through scenario-based testing that simulates real forensic cases in online exams. Finally, validation is conducted by consulting experts in the field, including digital forensic professionals and education technology specialists, to assess the relevance, completeness, and applicability of the framework in real-world settings.

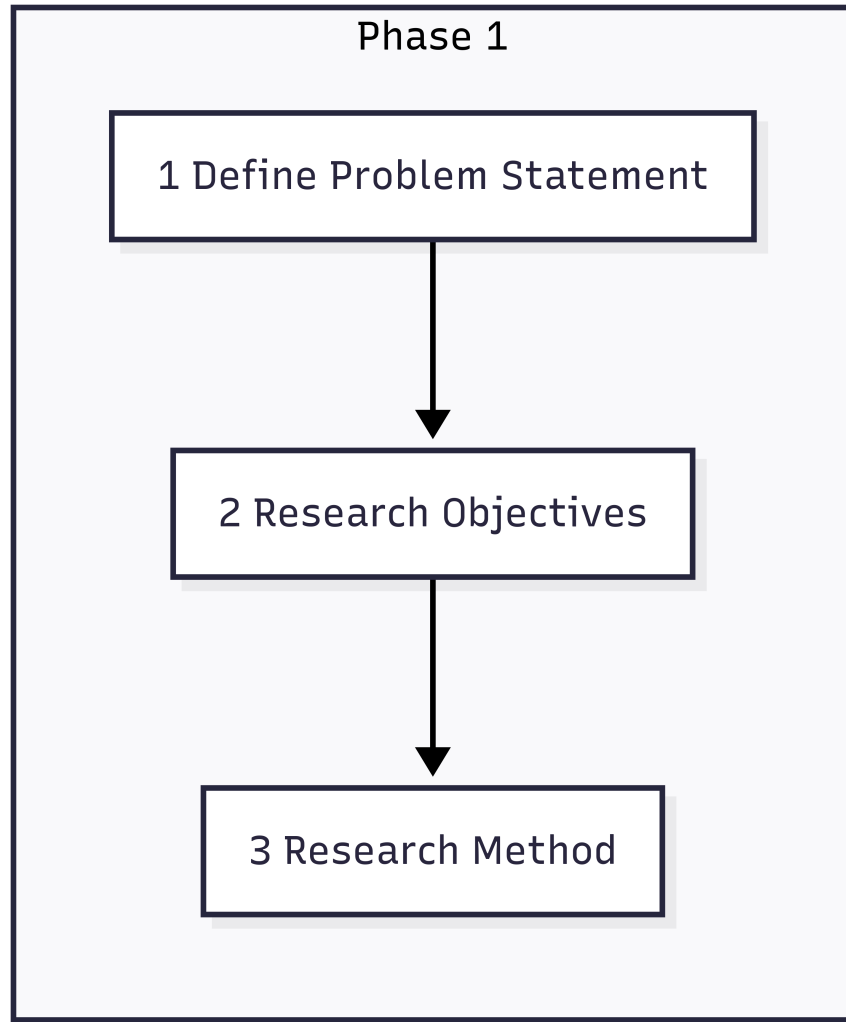


Figure 3.4: Proactive Forensic Framework Development Phase 1

As shown in Figure 3.4, the first phase focuses on understanding the problem domain by identifying the challenges associated with digital forensic investigations in online learning systems. This phase includes formulating the problem statement, defining research objectives, and selecting appropriate research methods. The output of this phase forms the foundation for all subsequent design and development activities.

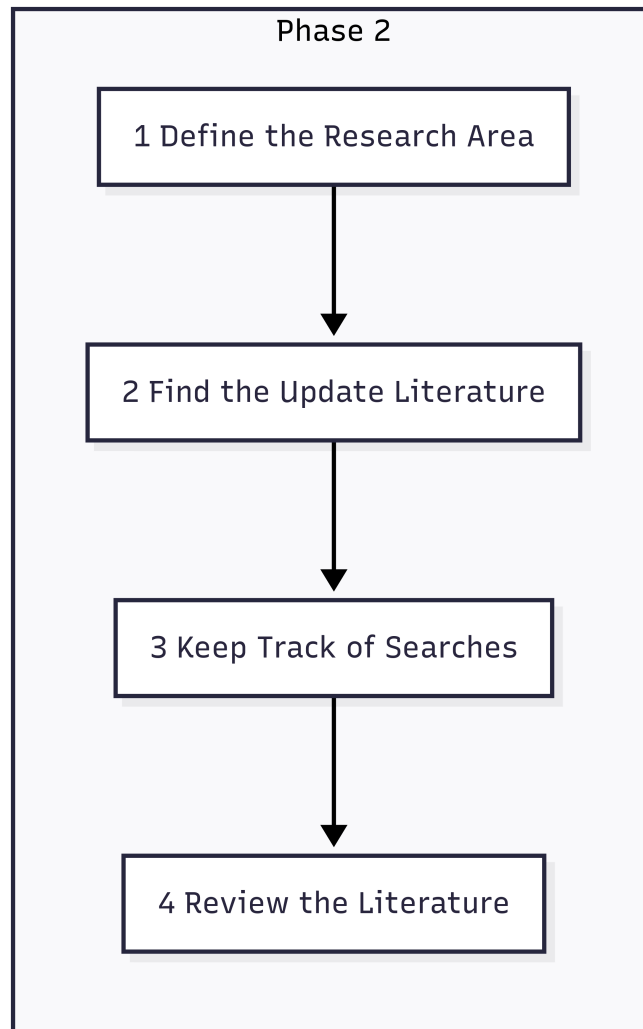


Figure 3.5: Proactive Forensic Framework Development Phase 2

Figure 3.5 presents Phase 2, which involves an extensive literature review to identify existing digital forensic frameworks, log management strategies, and relevant standards such as NIST SP 800-92. During this phase, the research area is scoped, updated literature is collected, search activity is recorded systematically, and selected works are critically analyzed. The aim is to extract best practices and determine the gaps the proposed framework should address.

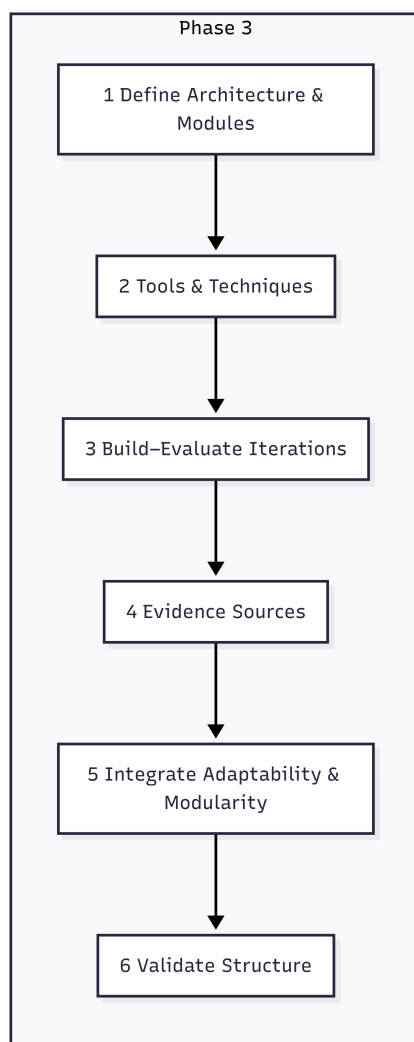


Figure 3.6: Proactive Forensic Framework Development Phase 3

In Phase 3, illustrated in Figure 3.6, the framework architecture is designed and adapted based on the needs identified in earlier phases. The design follows a modular approach, integrating components such as log identification, proactive collection, transmission, analysis, and reporting. Each module is specified to align with proactive forensics goals, including real-time monitoring and traceability.

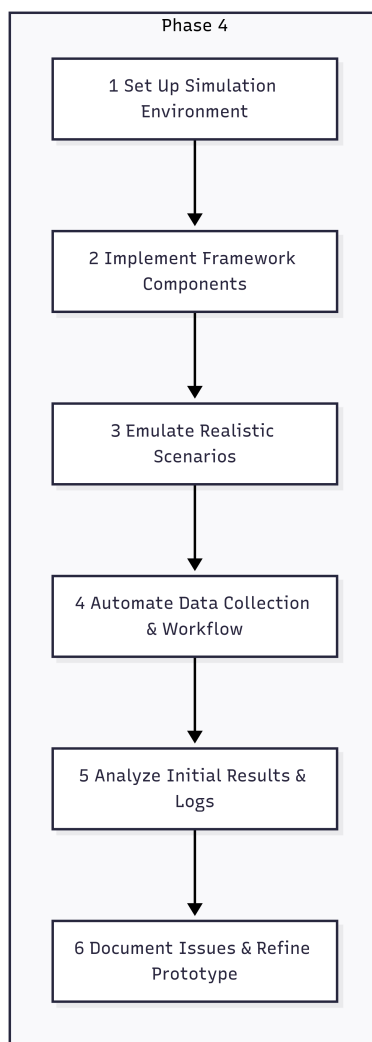


Figure 3.7: Proactive Forensic Framework Development Phase 4

Figure 3.7 details the implementation of the designed framework within a simulated environment. This phase includes configuring log sources, implementing data flow, integrating analysis tools, and simulating typical forensic scenarios (e.g., cheating attempts, unauthorized access). The prototype is developed iteratively, allowing for refinement based on observed results.

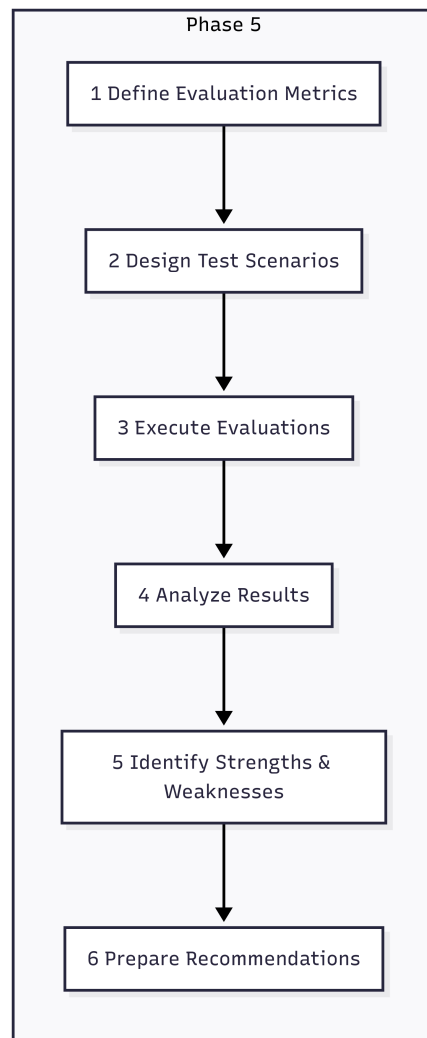


Figure 3.8: Proactive Forensic Framework Development Phase 5

Figure 3.8, focuses on evaluating the framework using predefined metrics such as accuracy, timeliness, and detection capability. Experts from the digital forensic domain are engaged to validate the framework through structured interviews and feedback instruments. Their input helps verify the practical relevance and completeness of the framework, ensuring its applicability in real-world online education settings.

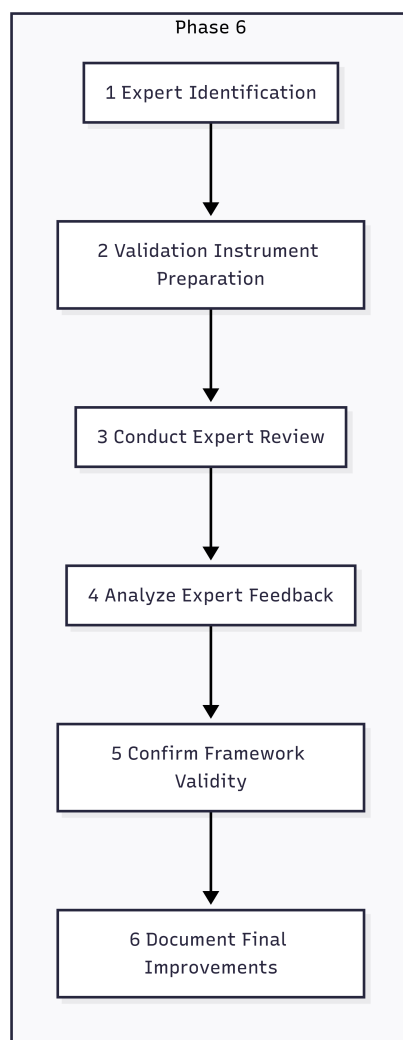


Figure 3.9: Proactive Forensic Framework Development Phase 6

The final phase is as shown in Figure 3.9, the sixth phase involves validating the overall framework through expert evaluation. This phase is crucial to ensure the scientific, technical, and practical credibility of the framework. The process begins with identifying suitable experts in digital forensics, cybersecurity, and online education systems. A structured validation instrument—such as a questionnaire or interview protocol—is developed to guide the expert review.

3.1.2 Log Management Implementation

The implementation of the log management system is designed to support proactive forensic readiness within an online examination environment. The system consists of several key components that are integrated to enable automated log acquisition, secure storage, and effective analysis. The implementation follows the NIST 800-92 framework as a reference for security log management practices.

The main components of the implementation are as follows:

- **Log Sources:** The system captures log data from multiple sources, including database logs (e.g., quiz attempts, login sessions), operating system logs, and application-level logs.
- **Log Collection:** A job scheduler is used to automate the periodic retrieval of log data. This ensures consistent and timely acquisition of relevant logs during and after exam sessions.
- **Log Transmission:** Collected logs are transmitted to a centralized storage server through secure channels. This step ensures all data is available for centralized processing and analysis.
- **Log Storage:** Logs are stored in a structured directory format with timestamping, access control, and checksum validation to ensure data integrity and traceability.
- **Log Analysis:** A dashboard interface is used to visualize log activity, while anomaly detection is performed using the Isolation Forest algorithm to identify suspicious patterns.
- **Alerting and Reporting:** The system provides real-time notifications for potential cheating incidents and generates structured PDF reports to support further forensic investigation.

This implementation ensures that log data is collected, preserved, and analyzed in a manner that supports digital forensic objectives, while also enabling timely administrative responses to security incidents during online examinations.

3.1.3 Log Management Testing Phase

The testing phase aims to ensure that each process within the implemented log management system functions correctly and supports proactive digital forensic readiness, particularly in the context of online examinations. The evaluation is conducted sequentially, based on the nine log management processes previously defined. Each stage is tested using scenarios that simulate realistic conditions and are aligned with modeled threats, such as impersonation attacks (e.g., exam proxy or “joki”) via remote access tools.

The testing stages are outlined as follows:

1. Log Identification Testing

Verifies that all log sources, including those from the exam platform, third-party applications, and operating systems, are successfully identified. The system should be capable of handling a variety of log formats, including those from legacy systems (e.g., plaintext, CSV).

2. Proactive Log Collection Testing

Ensures that log data is collected periodically through automated mechanisms. The system is tested under normal and high-load conditions to evaluate reliability and to identify potential performance bottlenecks.

3. Log Transmission Testing

Assesses the secure and timely transmission of log files from source systems to a centralized log server. The focus is on consistency, fault tolerance, and resistance to network delays or failures.

4. Log Storage Testing

Evaluates whether logs are stored in a structured and secure repository with appropriate access controls. Tests also verify retention policies and file integrity protection mechanisms.

5. Log Analyzer Testing

Confirms that the log analyzer or dashboard properly aggregates and displays data from various log formats. Usability and clarity of visualized user activity are key evaluation aspects.

6. Proactive Log Analysis Testing

Validates the system's capability to detect anomalies using machine learning techniques. Test data simulating both normal and suspicious behaviors is used to assess classification accuracy and anomaly detection effectiveness.

7. Notification Testing

Ensures that the system can generate and deliver alerts in response to detected suspicious activity. The testing includes verification of trigger conditions, content accuracy, and notification timeliness.

8. Log Preservation Testing

Verifies the system's ability to export and preserve logs in a tamper-evident format using cryptographic checksums. File integrity checks are conducted to confirm that no modifications occur after export.

9. Log Reporting Testing

Confirms the ability of the system to generate readable and standardized reports summarizing user activity, typically in PDF format. The report structure and content relevance are evaluated.

CHAPTER 4

PRESENTATION, ANALYSIS AND INTERPRETATION OF DATA

In thesis writing, the most difficult part to defend is chapter 4 because it is in this section where you will present the results of the whole study. Here is a sample thesis format.

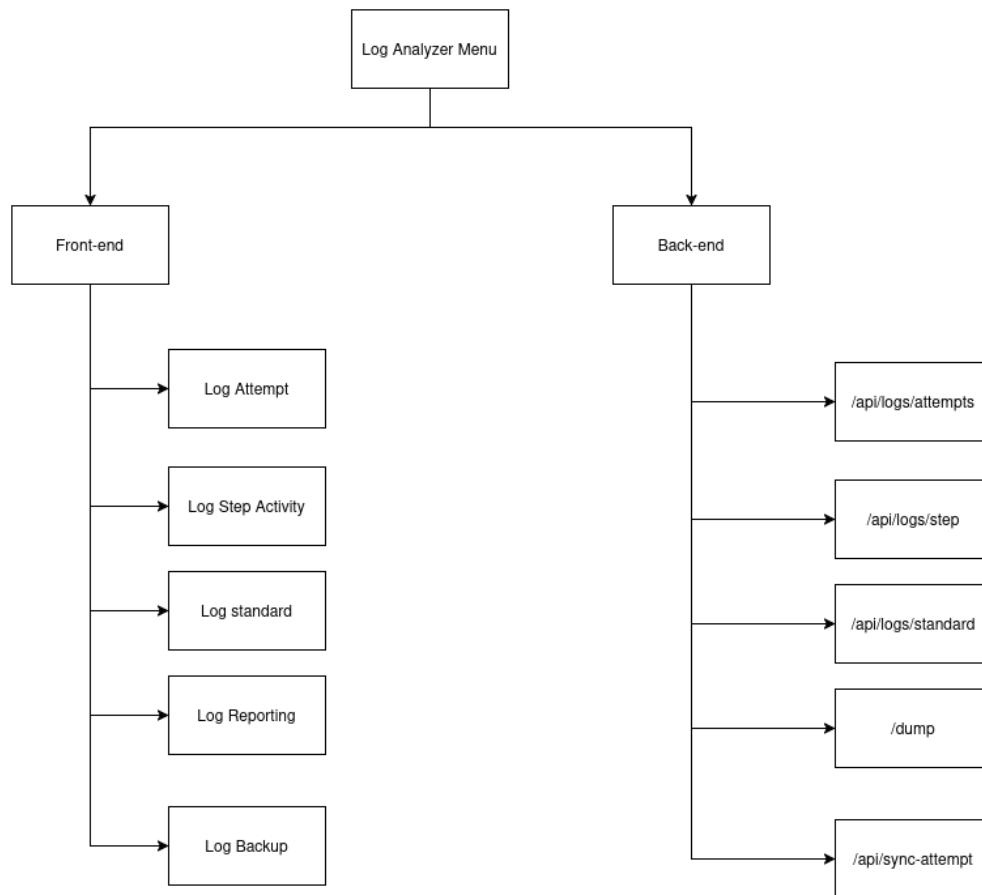


Figure 4.1: Log Management Structure Menu

The picture above 4.1 is a menu structure. the menu structure is used to display the structure of the log management application.

4.1 Phase 1

4.1.1 Define Problem Statement

- Server Attack
- Cheating Attack
- Software Attack (e.g., errors in the application)
- Client Device Attack
 - Network Device Attack
 - Client Device Attack
- Officer Attack (e.g., attack on proctor)
- Examinee Attack

List above Rosmansyah et al. [25] is attack-defence tree model systematically illustrates how various forms of attacks can occur in an online exam system and how defence measures can be integrated at each potential attack point. With this model, exam organizers can be better prepared to anticipate threats and ensure the continuity and security of the online examination process.

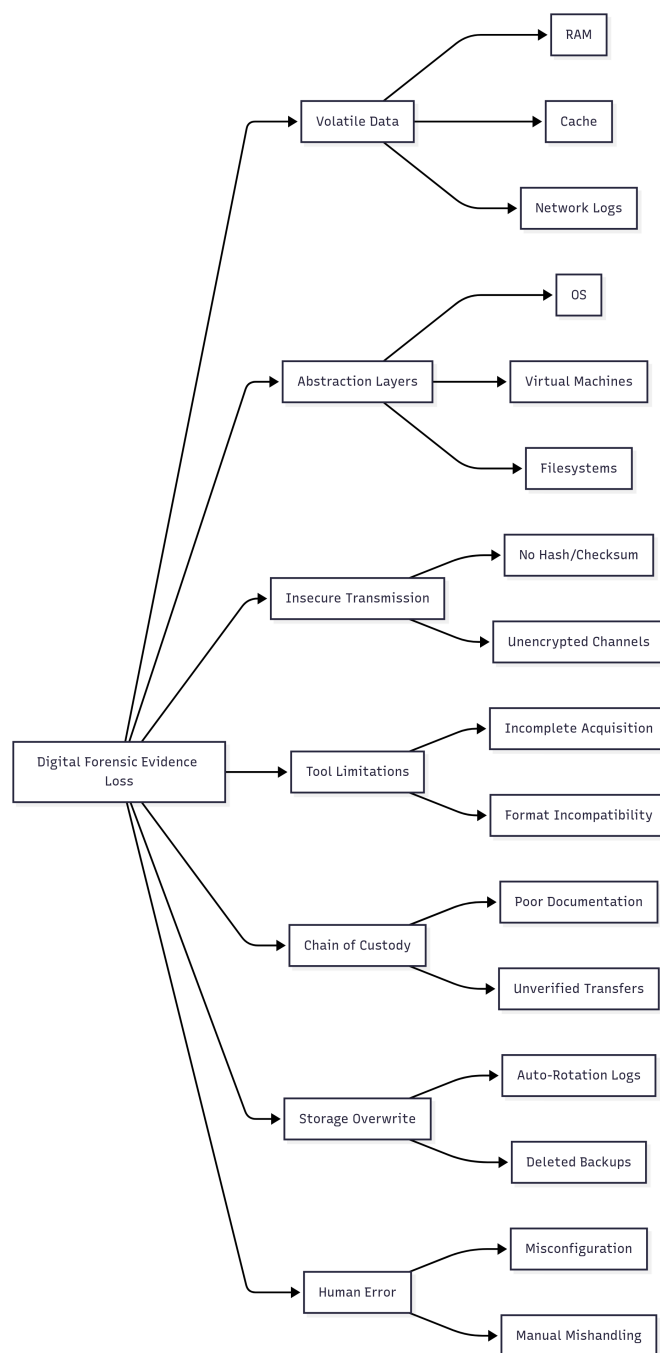


Figure 4.2: Digital Forensic Evidence Loss [17]

One of the major risks in digital forensic investigations is the loss of critical evidence due to various technical and procedural factors. These losses can compromise the integrity, completeness, and legal admissibility of evidence, thereby weakening the overall investigation. Evidence loss may occur during collection, transmission, storage, or analysis phases—often unnoticed until a post-incident review. Therefore, proactive mechanisms must be in place to preserve, validate, and monitor digital logs systematically.

Phase 1 of the proposed framework focuses on defining the core objectives and technical requirements, as introduced in Chapter 3. This phase emphasizes the proactive nature of digital forensic readiness, where logs are collected, preserved, and structured long before any examination session or security incident occurs.

This phase ensures that the framework's direction remains aligned with long-term forensic preparedness, enabling timely evidence collection and minimizing the risk of missing critical data when incidents do occur.

4.1.2 Research Objectives

The main objective is to build a centralized and automated log management system that enables continuous monitoring and log acquisition in an online learning environment. The system should not wait for anomalies or incidents to occur, but instead establish an audit trail that is always active and available for future analysis.

Key system requirements identified in this phase include:

- Continuous log acquisition from key components such as Moodle, the operating system, database services, and web servers.
- A proactive scheduling mechanism (e.g., `crontab`) to retrieve logs at defined intervals, regardless of exam schedules.
- Centralized log storage with access control, timestamping, and integrity verification using cryptographic hashes.
- Compatibility with scalable infrastructure (e.g., Azure VMSS) to support dynamic resource allocation.
- Capability to export logs in a structured format (CSV/JSON) for downstream forensic processes.
- Readiness for future integration with machine learning-based analysis.

4.1.3 Research Method

To achieve the objectives outlined in the previous section, a structured method was designed based on a system development approach, incorporating best practices from digital forensic readiness and proactive log management frameworks. The methodology integrates technical implementation, experimental testing, and expert validation to ensure both functional performance and forensic reliability.

The core steps of the method are as follows:

- **Framework Design:** A modular architecture is designed, comprising components such as a log collector, scheduler, centralized log storage, machine learning engine, dashboard interface,

and notification system. The design adheres to log lifecycle principles based on NIST SP 800-92.

- **Data Flow Implementation:** A data pipeline is developed to enable automated log acquisition, transmission using `rsync`, and storage into structured directories. The logs include user quiz attempts, question interaction steps, and system events from the Moodle platform.
- **Machine Learning Integration:** An unsupervised anomaly detection model (Isolation Forest) is trained using historical exam data to classify suspicious activity based on exam duration and scoring patterns.
- **Preservation and Reporting:** Log integrity is ensured through MD5 hashing and timestamping. Reports are generated in PDF format to assist investigators and administrators in analyzing flagged incidents.
- **Validation and Testing:** The system is tested in a simulated environment using real log data from an online English Proficiency Test (EPT), and the results are validated through expert review by digital forensic practitioners.

This method ensures that the framework is not only technically viable but also compliant with forensic principles of evidence preservation, integrity, and traceability. The combination of automation, machine learning, and structured log management offers a holistic solution for supporting proactive digital forensics in online examination systems.

4.2 Phase 2

Phase 2 focuses on conducting a comprehensive literature review to identify best practices, frameworks, and standards relevant to digital forensics in online education environments. This stage is critical for ensuring that the proposed framework is grounded in established knowledge and is capable of addressing current gaps in proactive forensic readiness.

The literature review emphasizes two main areas: (1) digital forensic frameworks, and (2) log management standards. Particular attention is given to the NIST Special Publication 800-92, which provides detailed guidelines for computer security log management. This standard outlines principles for log generation, collection, transmission, storage, analysis, and disposal, and serves as the core reference model for the framework in this study.

In addition, previous research by Adel et al [2]. and Alharbi et al[4]. was reviewed to understand approaches in proactive and reactive forensics, particularly in e-learning and cloud-based environments. These studies highlighted the importance of early evidence acquisition, modular framework design, and integration with real-world systems such as LMS and cloud platforms.

Key insights extracted from this phase include:

- The necessity of proactive log acquisition to minimize evidence loss.

- Integration of log lifecycle processes with operational systems.
- Modular design that allows future expansion into anomaly detection and incident response.
- Challenges in standardizing log formats across heterogeneous sources.

This literature review provided both theoretical and practical direction for designing a customized framework suitable for scalable, secure, and evidence-ready examination systems. It also helped define the scope and limitations to be addressed in the design phase that follows.

Table 4.1: Summary of Related Works and Relevance to This Research

Author(s)	Focus of Study	Relevance to Proposed Framework
Kent et al. (NIST SP 800-92, 2006) [13]	Guidelines for log generation, collection, transmission, storage, and disposal	Serves as the primary reference for structuring the proactive log management phases
Adel et al. (2021) [2]	Modular digital forensic framework for e-learning environments	Provides foundational model for framework phases, adapted and expanded in this study
Alharbi et al. (2020) [4]	Proactive and reactive digital forensic models in cloud-based systems	Supports the concept of readiness and proactive logging in scalable infrastructure
Smirani and Boulahia (2022) [28]	Evaluation metrics for machine learning in intrusion/anomaly detection	Guides the measurement of ML performance (accuracy, precision, recall, F1) in Phase 6
Garg and Goel (2023) Garg and Goel [9]	Application of ML for log evidence in academic integrity cases	Validates the use of Isolation Forest for detecting abnormal behavior in on-line exams

4.3 Phase 3

Phase 3 focuses on designing and customizing the forensic log management framework according to the objectives and insights gathered in the previous phases. This design stage emphasizes a modular, scalable, and proactive architecture aligned with the principles of NIST SP 800-92 and adapted from the framework structure of Adel et al.

The design process begins by mapping the nine essential log management processes—ranging from log identification to reporting—into functional components within the online examination ecosystem. Each component is defined to operate independently yet cohesively to support end-to-end forensic readiness.

Key modifications and design decisions include:

- **Modular architecture:** The framework is divided into clear modules such as log acquisition, transmission, storage, analysis, notification, and preservation. This allows independent testing and scaling.
- **Cloud deployment readiness:** The infrastructure is built using Azure Virtual Machine Scale Sets (VMSS) to support dynamic provisioning and load balancing during examination periods.
- **Integration with Moodle LMS:** The system is directly linked to Moodle’s internal logging and quiz attempt tracking mechanisms to enable real-time collection of learning and assessment activities.
- **Security and traceability:** The framework includes features for log timestamping, structured storage, and integrity verification using MD5 checksums to maintain evidential reliability.
- **Preparation for downstream analysis:** Logs are formatted in a machine-readable structure (CSV/JSON) to support future use in anomaly detection and forensic reporting.

Figure 4.21 illustrates the overall framework design adapted from NIST SP 800-92, enhanced to suit the specific requirements of online exam environments. This architecture ensures that the framework remains extensible and can be evaluated under simulated forensic scenarios in later phases.

4.4 Phase 4

Phase 4 involves the implementation and simulation of the designed forensic log management framework within a controlled test environment. This phase validates whether the architectural design and functional components can operate effectively in practice, especially in the context of online examination scenarios.

The simulation was conducted using a testbed environment that replicates the infrastructure of an actual university’s online exam system.

4.4.1 Log Identification

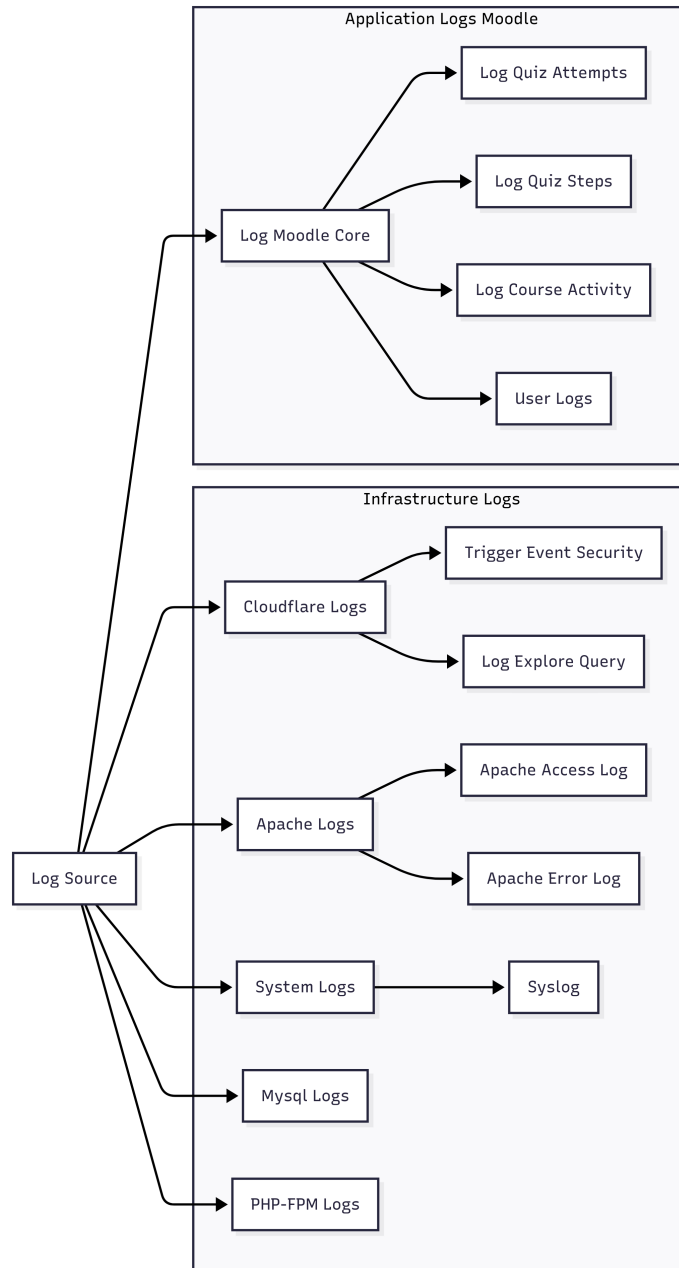


Figure 4.3: Log Source

The diagram 4.3 represents the result of an identification process of various log sources within the system environment. These sources are categorized into two main levels: **infrastructure level** and **application level**.

Table 4.2: Table Log Source, Description and Example Log Data

Log Source	Description	Example Log Data
Trigger Event Security	Log entries triggered by Cloudflare security rules (e.g. WAF, bot protection)	<pre> EventID: 38492 Type: SQL Injection Severity: High URI: /login/index.php </pre>
Log Explore Query	Cloudflare query logs from Log Explorer for audit and troubleshooting	<pre> query="SELECT * FROM traffic_logs WHERE status=403" </pre>
Apache Access Log	Log of all HTTP requests handled by Apache web server	<pre> 192.0.2.1 - - [21/ Jun /2025:10:04:32 +0000] "GET /moodle/login/ index.php HTTP /1.1" 200 4523 </pre>
Apache Error Log	Application/server-side error messages generated by Apache	<pre> [Sat Jun 21 10:05:12.123456 2025] [php:error] [client 192.0.2.1] PHP Fatal error: Call to undefined function... </pre>

Log Source	Description	Example Log Data
System Logs (Syslog)	General OS-level events such as service starts, reboots, errors	<pre>Feb 26 06:25:02 localhost rsyslogd: [origin software=" rsyslogd" swVersion ="8.32.0" x-pid="1332" x-info ="http://www. rsyslog.com"] rsyslogd was HUPed</pre>
MySQL Logs	Database-level warnings and errors from MySQL error log	<pre>2023-02-24T16 :27:52.714361Z 0 [Warning] Could not increase number of max_open_files to more than 5000 (request: 50000)</pre>
PHP-FPM Logs	Runtime messages related to PHP process management (PHP 7.2)	<pre>[08-Nov-2023 12:52:10] NOTICE: systemd monitor interval set to 10000ms</pre>

Log Source	Description	Example Log Data
Log Quiz Attempts	Records each attempt made by a user on a quiz including layout, timing, and score	<pre> attempt_id: 1028782 user_id: 34413 name: user course: EPT Home Edition quiz: Grammar uniqueid: 1033212 layout: 1,2,3,...,41,0 timestart: 1711939947 timefinish: 1711941217 score: 11 </pre>
Log Quiz Steps	Step-by-step interaction data per quiz question attempt, including timing and score	<pre> attempt_id: 1028782 user_id: 34413 quiz: Grammar question_attempt_id : 19613856 step_id: 64023090 step_state: todo step_start_time: 1711939947 next_step_time: 1711940179 time_spent_on_question : 232 </pre>

Log Source	Description	Example Log Data
Log Course Activity	Activity logs related to course views, resources accessed, etc.	<pre> Time: 23/06/25, 10:49 User full name: admin admin Event context: Course: EPrT HE Pre-Exam Event name: Course viewed IP address: 103.233.100.202 </pre>
User Logs	Tracks user actions across the platform (login, logout, enroll, etc.)	<pre> Time: 2025-06-21 09:55:00 User: student02 Action: loggedout </pre>

Log identification is done by identifying the sources that generate logs by reviewing the results from exam participants. The log attempt results will be stored in a database, therefore the log source is located in the MySQL database. In the process of log analysis related to online examinations in Moodle, several database tables have been identified as essential sources of information for understanding user behavior and quiz performance. These tables are involved in recording quiz attempts and linking them to user profiles, course structures, and module instances.

The key identified tables include:

- **mdl_quiz_attempts:** This table stores individual quiz attempt records by users. It contains timestamps for when an attempt started and finished, along with the score achieved and the state of the attempt (e.g., in progress, finished).
- **mdl_user:** This table contains user-specific information such as user ID, first name, last name, and authentication details. It allows quiz attempts to be linked directly to the participants.
- **mdl_quiz:** This table stores metadata about each quiz, including its name, settings, grading method, and association to a course.
- **mdl_course:** This table defines course-level information. It helps in organizing and grouping quizzes under the appropriate academic or training program context.

- **mdl_course_modules**: This table links quizzes to specific module instances within a course, enabling fine-grained selection of quiz activity based on module ID. It is critical for identifying which quizzes are deployed in which course sections.

Together, these tables provide a comprehensive relational schema for extracting, preprocessing, and analyzing quiz attempt data in a structured and meaningful way.

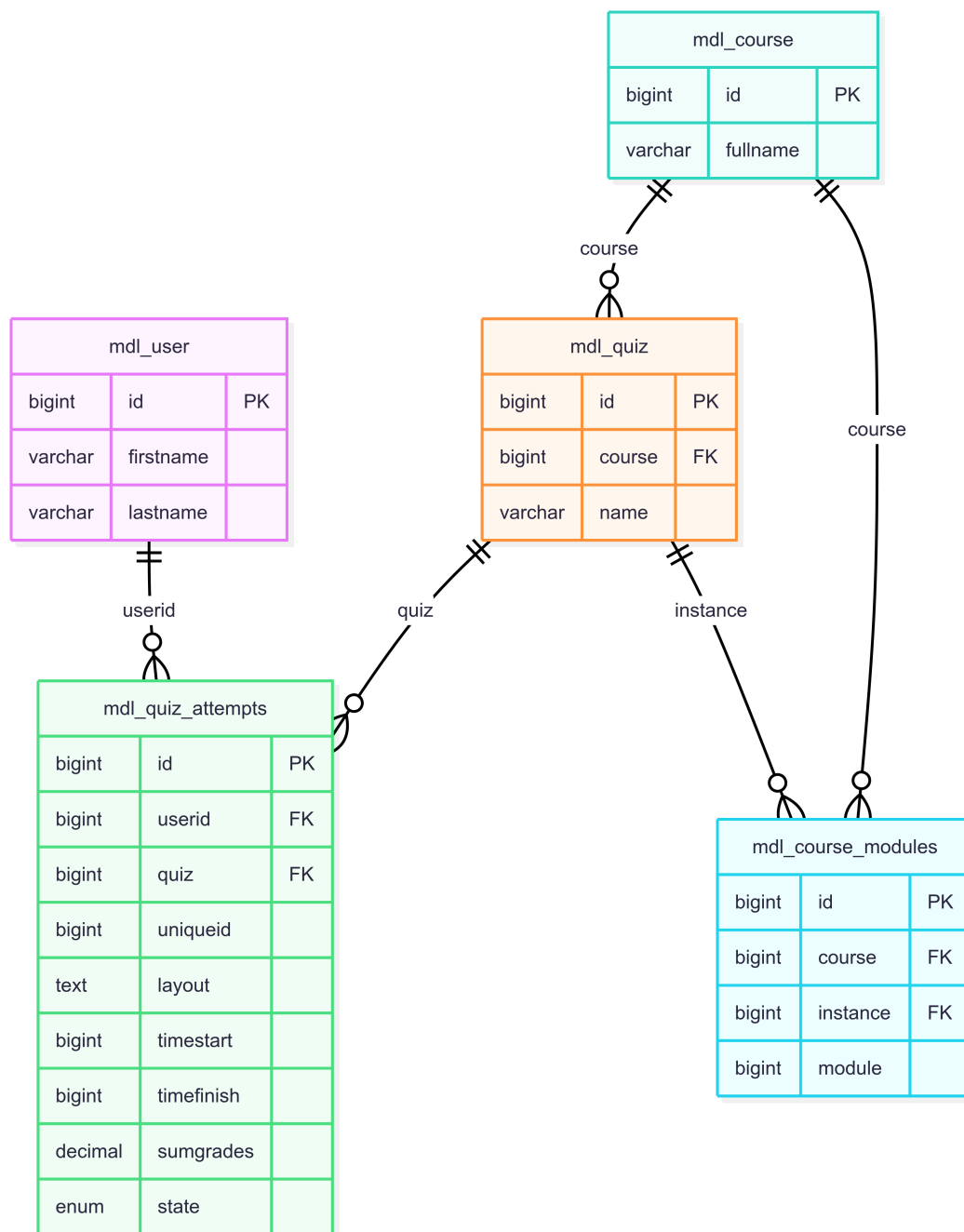


Figure 4.4: ERD from Several Column Tables

The Entity Relationship Diagram (ERD) presented above illustrates the core database schema used to represent online exam results within the Moodle learning management system.

Index of /laclog/lac-eprt-log





<u>Name</u>	<u>Last modified</u>	<u>Size</u> <u>Description</u>
 Parent Directory		-
 Log Course Activity/	2024-04-05 10:01	-
 Quiz Attempts/	2024-04-05 10:01	-
 Quiz Attempts Step/	2024-04-05 10:01	-

Figure 4.5: Backup from mysql dump

Fig 4.5 is the quiz attempts of exam participants. This data will be utilized for analyzing the behavior of the exam participants.

Table 4.3: Course Log Records from User Attempts Quiz

Attempt ID	User ID	Name	Course	Quiz	Layout	Start Time	Finish Time	Score
1028782	34413	Anon	EPrT Home Edition	Grammar	41 items	1711939947	1711941217	11
1028779	33532	Anon	EPrT Home Edition	Grammar	41 items	1711939560	1711940973	18
1028776	33601	Anon	EPrT Home Edition	Grammar	41 items	1711939379	1711940864	13
1028773	33718	Anon	EPrT Home Edition	Grammar	41 items	1711939355	1711940774	29

Table 4.4: Course Log Records from Moodle Course

Time	User full name	Affected user	Event text	Component	Event name	Description	Origin	IP address
23/06/25, 10:51	admin admin	-	Course: EPrT HE Pre-Exam	Logs	Log report viewed	The user with id '4' viewed the log report for the course with id '4'.	web	103.233.100
23/06/25, 10:51	admin admin	-	Course: EPrT HE Pre-Exam	Logs	Log report viewed	The user with id '4' viewed the log report for the course with id '4'.	web	103.233.100
23/06/25, 10:51	admin admin	-	Course: EPrT HE Pre-Exam	Logs	Log report viewed	The user with id '4' viewed the log report for the course with id '4'.	web	103.233.100
23/06/25, 10:49	admin admin	-	Course: EPrT HE Pre-Exam	System	Course viewed	The user with id '4' viewed the course with id '4'.	web	103.233.100

Fig 4.9 is logs data quiz attempts. By analyzing the collected data, valuable insights can be gained into the behavior of users during exams. One such insight can be gleaned from the time participants take to complete the exam.

4.4.2 Proactive Collection

In modern system administration, the use of **cron job** on Linux/Unix systems is a prevalent method for performing automated, periodic tasks such as log extraction, data transformation, or backup. Cron enables precise scheduling through the **crontab** utility, which defines job frequency using five time fields: minute, hour, day, month, and weekday Davidovič and Guliani [7].

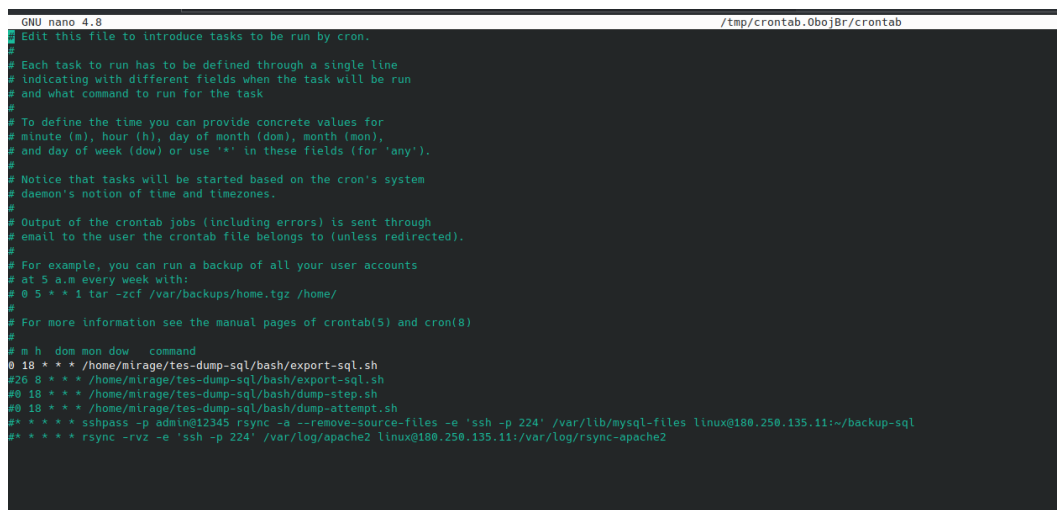
Academic validation of this method is found in related research. For instance, Hazwam et al. propose a cron-triggered Perl script to periodically parse honeypot logs into a database, significantly reducing storage requirements and improving system performance Halim et al. [10]. This demonstrates the effectiveness of cron-based scheduling for resource-efficient log handling.

Furthermore, system reliability and auditing are supported by best practices from NIST SP 800-92, which emphasize the scheduling of log management routines for integrity checks and retention.

In summary, **cron-based periodic log collection** offers a lightweight, scalable, and verifiable method for preserving and processing database-derived logs, and is validated both in the field and in academic literature.

Proactive collection for automated collection. automated collection systematic for collecting and storing data or evidence before incidents occur that could lead to data loss. There are three types of logs that will be collected: attempt logs, quiz attempt step logs, and course activity logs.

```
0 18 * * * * /home/mirage/tes-dump-sql/export-sql.sh
```



```
GNU nano 4.8 /tmp/crontab.0bojBr/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# e h dom mon dow   command
0 18 * * * /home/mirage/tes-dump-sql/bash/export-sql.sh
0 26 8 * * * /home/mirage/tes-dump-sql/bash/export-sql.sh
0 0 18 * * * /home/mirage/tes-dump-sql/bash/dump-step.sh
0 0 18 * * * /home/mirage/tes-dump-sql/bash/dump-attempt.sh
* * * * * sshpass -p admin@12345 rsync -a --remove-source-files -e 'ssh -p 224' /var/lib/mysql-files linux@180.250.135.11:~/backup-sql
* * * * * rsync -rvz -e 'ssh -p 224' /var/log/apache2 linux@180.250.135.11:/var/log/rsync-apache2
```

Figure 4.6: Scheduler for proactive collection

Figure 4.6 shows the job scheduler used for periodic log collection. It automates the retrieval of log data from the database and system components, ensuring timely collection and synchronization to centralized storage to support proactive forensic. To extract detailed quiz attempt logs for further analysis, a structured SQL query was designed to retrieve records from multiple interconnected tables within the Moodle database schema. The focus of this query is to obtain finished quiz attempts from specific course modules associated with the quiz activity.

To facilitate log analysis related to online examinations within the Moodle platform, a specific SQL query was constructed to extract quiz attempt data based on the associated course identifier (course id). This approach ensures that only relevant records tied to a particular course and quiz activity are processed, improving both performance and precision in downstream analytics.

4.4.3 Log Transmission

To support secure and efficient log transmission, the system utilizes the **rsync** protocol to transfer collected log files from the master virtual machine (VM), which hosts the examination platform, to a centralized log storage server. The use of **rsync** enables incremental synchronization, ensuring that only updated or newly generated log data is transmitted, thereby reducing bandwidth usage and improving transfer speed. This transmission occurs as part of a scheduled routine executed daily after examination sessions conclude, ensuring timely backup and availability of log data for further analysis and archival.

Rsync was selected over SCP for log transmission because it supports delta transfer, checksum verification, and transfer resumption—features that ensure both performance and integrity during multiple, incremental log backups. In contrast, SCP lacks these forensic-grade guarantees [20, 30, 31].

```
#!/bin/bash
set -a
source "${dirname "$0"}/.env"
set +a

min=$(date +%Y-%m-%d-%T)
fullpath=/var/lib/mysql-files/$min

mysql -u root -padmin@123 moodle < ../query-sql/log.sql
mysql -u root -padmin@123 moodle < ../query-sql/step.sql
mysql -u root -padmin@123 moodle < ../query-sql/attempt.sql

mkdir $fullpath
mv /var/lib/mysql-files/*.csv $fullpath
echo $fullpath
sleep 5
sshpass -p "$PASSWORD" rsync -a --remove-source-files -e "ssh -p $PORT" /var/lib/mysql-files/$min "$REMOTE_USER"@"$REMOTE_HOST":~/backup-sql
```

Figure 4.7: Exporter script

A scheduled task is configured using the Linux **crontab** utility to automate daily log collection at 18:00 local time. This ensures logs are consistently backed up at the end of each examination session. exporter software component used to collect and transfer data from one system or virtual machine (VM) to another. By using exporters, organizations can automate data workflows, centralize information, and gain valuable insights from their systems. Whether for monitoring, logging,

or data migration.

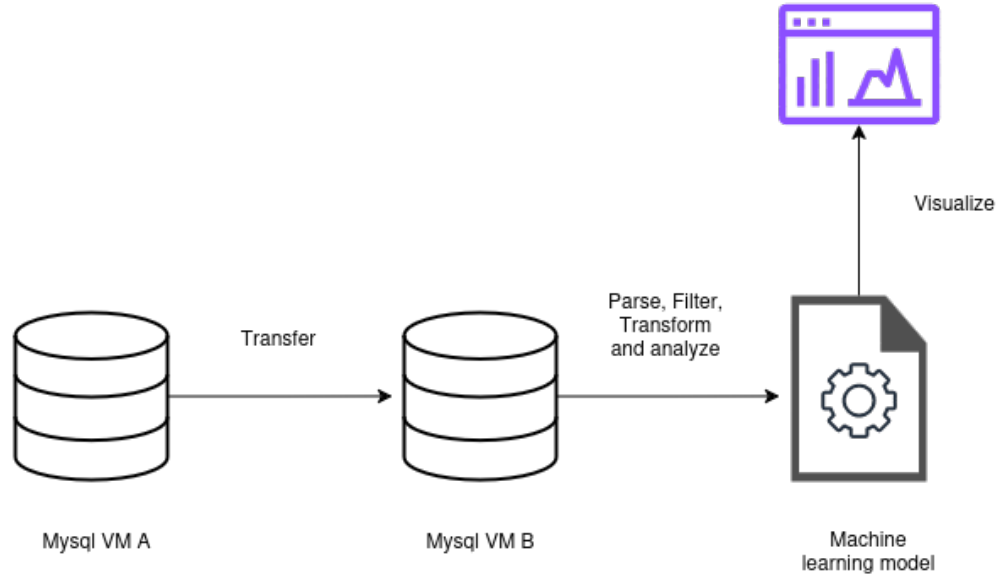


Figure 4.8: Machine learning workflow

Machine learning workflow 4.8 which will read from the vm-b database. Data from the vm-b database will be processed by the machine learning model that has been created.

4.4.4 Log Storage

In the context of proactive forensic log management, the log storage component plays a critical role in ensuring data availability, integrity, and traceability over time. Logs that have been proactively collected and transmitted must be preserved in a structured and secure manner to support future investigations, audits, and potential legal proceedings.

```

root@slave-db:/home/linux/backup-sql# ls
2024-09-04-21:35:05 2024-09-08-01:00:01 2024-09-10-01:00:01 2024-09-22-01:00:01 2024-10-19-01:00:01 2024-11-15-01:00:01 2024-12-12-01:00:01
2024-09-04-21:50:55 2024-09-09-01:00:02 2024-09-10-01:01:41 2024-09-23-01:00:01 2024-10-20-01:00:01 2024-11-16-01:00:01 2024-12-13-01:00:01
2024-09-04-21:57:47 2024-09-09-13:50:10 2024-09-10-01:03:07 2024-09-24-01:00:01 2024-10-21-01:00:01 2024-11-17-01:00:01 2024-12-15-01:00:01
2024-09-04-22:00:47 2024-09-09-14:01:01 2024-09-10-01:03:26 2024-09-25-01:00:01 2024-10-22-01:00:01 2024-11-18-01:00:01 2024-12-16-01:00:01
2024-09-04-22:11:02 2024-09-09-14:02:18 2024-09-10-01:03:45 2024-09-26-01:00:01 2024-10-23-01:00:01 2024-11-19-01:00:01 2024-12-17-01:00:01
2024-09-05-04:23:01 2024-09-09-14:46:16 2024-09-10-01:09:50 2024-09-27-01:00:01 2024-10-24-01:00:01 2024-11-20-01:00:01 2024-12-18-01:00:01
2024-09-05-11:57:07 2024-09-09-14:46:18 2024-09-10-01:10:26 2024-09-28-01:00:01 2024-10-25-01:00:01 2024-11-21-01:00:01 2024-12-19-01:00:01
2024-09-05-11:59:58 2024-09-09-14:50:55 2024-09-10-01:11:01 2024-09-29-01:00:01 2024-10-26-01:00:01 2024-11-22-01:00:01 2024-12-20-01:00:01
2024-09-05-12:03:05 2024-09-09-16:03:56 2024-09-10-01:11:41 2024-09-30-01:00:01 2024-10-27-01:00:01 2024-11-23-01:00:01 2024-12-21-01:00:01
2024-09-05-12:06:28 2024-09-09-16:04:00 2024-09-10-09:36:36 2024-10-01-01:00:01 2024-10-28-01:00:01 2024-11-24-01:00:02 2024-12-22-01:00:01
2024-09-05-12:09:33 2024-09-09-16:04:34 2024-09-10-13:39:55 2024-10-02-01:00:01 2024-10-29-01:00:01 2024-11-25-01:00:01 2024-12-23-01:00:01
2024-09-05-12:12:50 2024-09-09-16:36:04 2024-09-10-22:18:32 2024-10-03-01:00:01 2024-10-30-01:00:01 2024-11-26-01:00:01 2024-12-24-01:00:01
2024-09-05-12:14:54 2024-09-09-16:46:22 2024-09-10-22:20:01 2024-10-04-01:00:01 2024-10-31-01:00:01 2024-11-27-01:00:01 2024-12-25-01:00:01
2024-09-05-12:16:51 2024-09-09-16:47:02 2024-09-10-22:20:08 2024-10-05-01:00:01 2024-11-01-01:00:01 2024-11-28-01:00:01 2024-12-26-01:00:01
2024-09-05-12:19:34 2024-09-09-16:50:57 2024-09-10-22:21:10 2024-10-06-01:00:01 2024-11-02-01:00:01 2024-11-29-01:00:01 2024-12-27-01:00:01
2024-09-05-12:21:21 2024-09-09-17:08:33 2024-09-10-22:22:34 2024-10-07-01:00:01 2024-11-03-01:00:01 2024-11-30-01:00:01 2024-12-28-01:00:01
2024-09-05-12:36:34 2024-09-09-17:24:51 2024-09-11-01:00:01 2024-10-08-01:00:01 2024-11-04-01:00:01 2024-12-01-01:00:01 2024-12-29-01:00:01
2024-09-05-12:38:37 2024-09-09-17:34:08 2024-09-12-01:00:01 2024-10-09-01:00:01 2024-11-05-01:00:01 2024-12-02-01:00:01 2024-12-30-01:00:01
2024-09-05-12:40:50 2024-09-10-00:05:40 2024-09-13-01:00:01 2024-10-10-01:00:01 2024-11-06-01:00:01 2024-12-03-01:00:01 2024-12-31-01:00:01
2024-09-05-15:48:53 2024-09-10-00:07:20 2024-09-14-01:00:01 2024-10-11-01:00:01 2024-11-07-01:00:01 2024-12-04-01:00:01 2025-01-01-01:00:01
2024-09-05-15:49:42 2024-09-10-00:07:48 2024-09-15-01:00:01 2024-10-12-01:00:01 2024-11-08-01:00:01 2024-12-05-01:00:01 2025-01-02-01:00:01
2024-09-05-15:51:16 2024-09-10-00:08:20 2024-09-16-01:00:01 2024-10-13-01:00:01 2024-11-09-01:00:01 2024-12-06-01:00:01 2025-01-03-01:00:02
2024-09-05-16:12:13 2024-09-10-00:09:19 2024-09-17-01:00:01 2024-10-14-01:00:01 2024-11-10-01:00:01 2024-12-07-01:00:01 2025-01-04-01:00:01
2024-09-05-21:25:48 2024-09-10-00:28:33 2024-09-18-01:00:01 2024-10-15-01:00:01 2024-11-11-01:00:01 2024-12-08-01:00:01 2025-01-05-01:00:01
2024-09-05-21:26:41 2024-09-10-00:30:16 2024-09-19-01:00:01 2024-10-16-01:00:01 2024-11-12-01:00:01 2024-12-09-01:00:01 2025-01-06-01:00:01
2024-09-06-04:23:01 2024-09-10-00:31:14 2024-09-20-01:00:01 2024-10-17-01:00:01 2024-11-13-01:00:01 2024-12-10-01:00:01 2025-01-07-01:00:01
2024-09-07-04:23:01 2024-09-10-00:51:50 2024-09-21-01:00:01 2024-10-18-01:00:01 2024-11-14-01:00:01 2024-12-11-01:00:01 mysql-files
root@slave-db:/home/linux/backup-sql#

```

Figure 4.9: Log data backup

Figure 4.9 To ensure efficient log management and prevent data loss, user activity data from the MySQL database related to the English Proficiency Test (EPT) is collected daily and stored in a centralized system. Each log entry is timestamped for accurate tracking, while strategies like log rotation, compression, and regular backups safeguard against data loss. This streamlined approach optimizes storage, enhances accessibility, and ensures the availability of critical data for forensic analysis.

4.4.5 Analysis of the Data

The data collection process has been completed. The next step involves analyzing the gathered data, specifically the data of exam participants who have attempted the EPRT quiz. Two key indicators will be considered:

Column Name	Example of Data
attempt_id	1029126
id	76022
firstname	NURUL
lastname	IZZAH LUTHFIAH NUR
course_name	EPrT Home Edition
quiz_name	Grammar
uniqueid	1033556
layout	1,0,2,3,4,5,6,0,7,8,9,10,11,0,12,13,14,15,16,0...
timestart	2024-04-02 07:39:53
timefinish	2024-04-02 08:04:29
score	26
diff_time	0 days 00:24:36
diff_time_minute	24.600000
epoch_start	1712043593
epoch_finish	1712045069
time_diff_seconds	1476
anomaly	1

Table 4.5: Example of column names and corresponding data in the online exam anomaly detection system

Table 4.5 presents a sample of the data used in the anomaly detection system for online examinations utilizing the *Isolation Forest* algorithm. The dataset consists of various attributes that represent the participants' activity during the exam, ranging from identity information to time-based metrics and performance results.

Key attributes used include:

- **attempt_id**, **id**, and **uniqueid**: Unique identifiers for each exam session.
- **firstname** and **lastname**: The name of the exam participant.
- **course_name** and **quiz_name**: Indicate the course title and the type of quiz taken.
- **layout**: Stores the sequence of questions accessed by the participant during the exam.
- **timestart** and **timefinish**: Represent the start and end timestamps of the exam attempt.

- **diff_time**, **diff_time_minute**, and **time_diff_seconds**: Metrics that capture the total time taken to complete the exam in various units.
- **score**: The final score obtained by the participant on the quiz.
- **anomaly**: The result of anomaly detection, where a value of 1 indicates that the data is classified as anomalous, and 0 means it is considered normal.

Table 4.6: Features Used for Analysis Data

Feature	Details	Data Value
Time_taken	Time taken by examinee to finish the session by calculating the difference of timestamp between examinee's start time and finish time	00:19:14
score	Score of an examinee in a session, this represents how many questions the examinee answered correctly	22

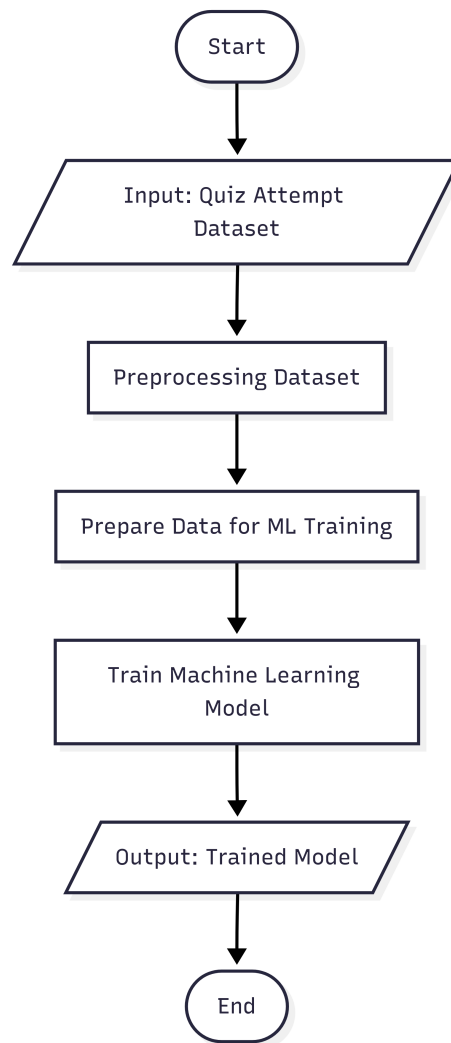


Figure 4.10: Flowchart ML Training

Figure 4.10 illustrates the overall process involved in training a machine learning (ML) model for proactive forensic analysis. The flowchart begins with the acquisition of training data, typically in the form of log files or structured events collected during simulated or real-world scenarios. These datasets undergo preprocessing steps, which may include cleaning, normalization, and feature extraction to ensure they are suitable for model input.

The machine learning model was trained using a dataset comprising 223 labeled instances. To ensure reliable evaluation and prevent class imbalance, the dataset was partitioned using an 80:20 split ratio with stratification. Specifically, 178 instances were allocated for training (`X_train`, `y_train`), and 45 instances for testing (`X_test`, `y_test`). The use of `stratify=y` ensured that the class distribution in both training and testing subsets remained proportional to the original dataset.

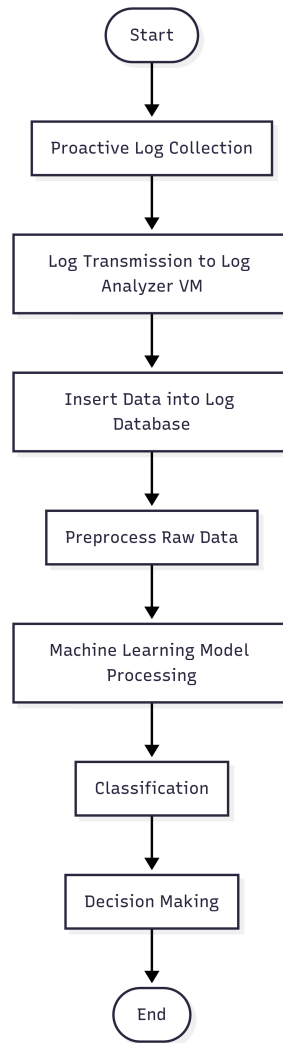


Figure 4.11: Flowchart Log Analysis

In previous research using machine learning to analyze the data log evidence in online exam Garg and Goel [9]. Furthermore, to tackle the ongoing challenge of establishing the ground truth in cases of academic dishonesty.

An anomaly detection model was developed using the Isolation Forest algorithm, which is well-suited for identifying outliers in high-dimensional datasets. The model was configured with a contamination rate of 0.005, indicating that approximately 0.5% of the data is assumed to be anomalous. This low contamination value reflects the expectation that only a very small portion of the dataset represents abnormal behavior.

To enhance the robustness of anomaly detection, the model utilizes 200 estimators (`n_estimators`), meaning that 200 isolation trees were built during the training process. The `max_samples` parameter was set to 0.8, allowing each tree to be trained on 80% of the available data, which introduces diversity among trees and improves generalization. Additionally, the model uses `max_features` set to 0.75, meaning that only 75% of the total features are considered when constructing each tree,

further increasing randomness and reducing overfitting.

Finally, the random state was fixed at 42 to ensure reproducibility across multiple training sessions. This configuration balances sensitivity to rare anomalies with model stability, making it suitable for detecting suspicious behavior in forensic log data.

Use of unsupervised algorithms because the dataset used is unlabeled. The dataset is taken from campuses located in Indonesia with online exams. The dataset used for training machine learning from the exam results on April 1 to 5, 2023. Number of datasets used 223.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

Accuracy is the percentage of correct predictions that a learner has achieved. It is computed by dividing the number of correct estimates by the total number of prediction Smirani and Boulahia [28].

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4.2)$$

Precision also known as the positive predictive value, is the ratio of the pertinent instances to the retrieved instances Smirani and Boulahia [28].

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4.3)$$

Recall is also called sensitivity, is a fragment of the retrieved relevant instance Smirani and Boulahia [28].

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.4)$$

F1-Score is a statistical measure that combines precision and recall with rate performance Smirani and Boulahia [28].

Table 4.7: Classification Report with Precision, Recall, and F1-score

Class	Precision	Recall	F1-score	Support
Dishonest	0.50	0.33	0.40	39
Honest	0.87	0.93	0.90	184
Accuracy	0.83			
Macro Avg	0.68	0.63	0.65	223
Weighted Avg	0.80	0.83	0.81	223

Table 4.7 is the result of evaluating the classification model using Precision, Recall, F1-score, and Support metrics. This model is most likely to be used for anomaly detection. The specified limit or threshold, such as scores below 40.

4.4.6 Log Monitoring

In this research, a custom-developed log management dashboard was implemented to support the forensic readiness framework. Unlike commercial Security Information and Event Management (SIEM) solutions such as Splunk, IBM QRadar, or Elastic Stack (ELK), the custom dashboard was chosen due to its alignment with the goals of flexibility, lightweight deployment, educational accessibility, and forensic specificity. **Key reasons for using a custom dashboard include:**

- **Cost-efficiency:** SIEM platforms often require commercial licenses or high infrastructure costs, which are not feasible for academic environments or small institutions. A custom solution removes this barrier [5].
- **Forensic Tailoring:** The custom dashboard was purpose-built to handle logs relevant to online examination systems, such as Moodle quiz attempts and user activity, which may not be natively supported or easily modeled in general-purpose SIEMs.
- **Transparency and Control:** Full access to the dashboard's source code and data pipelines allows greater transparency in log handling, which is critical for forensic validation and legal defensibility.
- **Lightweight and Focused:** SIEM tools often include broad and heavy telemetry modules not needed in this research. The custom dashboard uses a minimal stack (e.g., Flask, SQLite, or REST API) optimized for educational testing environments.
- **Ease of Integration with ML Models:** Integrating a machine learning-based anomaly detection model (e.g., Isolation Forest) directly into the custom dashboard is more straightforward than embedding it into a complex SIEM architecture.
- **Educational and Experimental Use:** In academic research, developing a tailored system allows hands-on experimentation with log structures, forensic workflows, and UI/UX designs—something not always possible with closed or semi-closed SIEMs.

Based on these reasons, a custom dashboard provides an ideal platform for validating forensic concepts in a resource-constrained environment while still ensuring critical functionality such as log analysis, visualization, classification, and anomaly alerting.

User ID	Firstname	Attempt ID	Course Name	Question Attempt ID	Quiz Name	Step State
2	moodiedude	25	EPIT HE Pre-Exam	129	Listening Pre-Exam	TO DO
2	moodiedude	25	EPIT HE Pre-Exam	129	Listening Pre-Exam	COMPLETE
2	moodiedude	25	EPIT HE Pre-Exam	129	Listening Pre-Exam	GRADE RIGHT
2	moodiedude	25	EPIT HE Pre-Exam	130	Listening Pre-Exam	TO DO
2	moodiedude	25	EPIT HE Pre-Exam	130	Listening Pre-Exam	COMPLETE
2	moodiedude	25	EPIT HE Pre-Exam	130	Listening Pre-Exam	GRADE WRONG
2	moodiedude	25	EPIT HE Pre-Exam	131	Listening Pre-Exam	TO DO
2	moodiedude	25	EPIT HE Pre-Exam	131	Listening Pre-Exam	COMPLETE
2	moodiedude	25	EPIT HE Pre-Exam	131	Listening Pre-Exam	GRADE WRONG
2	moodiedude	25	EPIT HE Pre-Exam	132	Listening Pre-Exam	TO DO

Figure 4.12: Log Attempt Step

Figure 4.12 illustrates the sequential log attempt steps recorded during an online examination session. Each log entry corresponds to a specific user action captured by the system and is categorized based on its execution state. These states represent the progression and outcome of each exam interaction, allowing forensic analysis to reconstruct user behavior.

The main states observed in the figure include **todo**, which signifies that the question was displayed to the participant but has not been answered; **complete**, indicating the participant submitted an answer; and two graded states: **gradedwrong** and **gradedright**, which show the automatic evaluation outcome of the submitted response. These log steps are timestamped and ordered, providing a temporal context for each transition.

User ID	Username	Full Name	Time Start	Time Finished	Time Date	Durasi Pengerjaan	Nilai	Status
36495	MUTHIA	NURHIKMAH	14:06:53	14:51:34	Mon, 01 Apr 2024 07:06:53 GMT	00:44:41	23	TERINDIKASI
106222	ILHAM	AGUSTIAN	14:06:55	14:36:54	Mon, 01 Apr 2024 07:06:55 GMT	00:29:59	44	TERINDIKASI
106222	ILHAM	AGUSTIAN	14:06:55	14:36:54	Mon, 01 Apr 2024 07:06:55 GMT	00:29:59	44	TERINDIKASI
36495	MUTHIA	NURHIKMAH	14:06:53	14:51:34	Mon, 01 Apr 2024 07:06:53 GMT	00:44:41	23	TERINDIKASI
35230	ROUDLOTUL	JANNAH	14:06:56	14:42:09	Mon, 01 Apr 2024 07:06:56 GMT	00:35:13	20	TERINDIKASI
36495	MUTHIA	NURHIKMAH	14:06:53	14:51:34	Mon, 01 Apr 2024 07:06:53 GMT	00:44:41	23	TERINDIKASI

Figure 4.13: Finding user

The figure 4.13 to find case with user to indication cheating. On this page, there are users who are suspected of cheating. The data obtained will be stored in a database, including the timestamp

and information on when the user committed the cheating. Therefore, the proctor will conduct another check after the exam.

4.4.7 Notification

In the development of the forensic log management framework, Telegram was selected as the primary alerting mechanism for several practical and technical reasons. While various alternatives exist for real-time notification systems such as email, Slack, Microsoft Teams, or SIEM integrated alerting the use of the Telegram Bot API offers a unique combination of simplicity, cost-effectiveness, and flexibility suitable for research and institutional deployments [34].

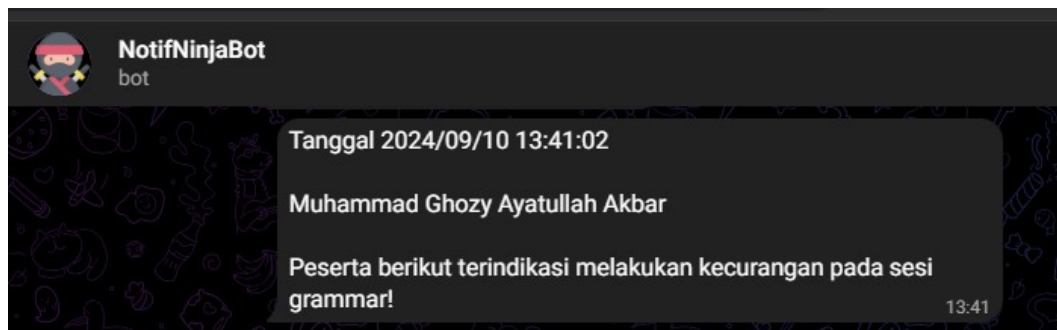
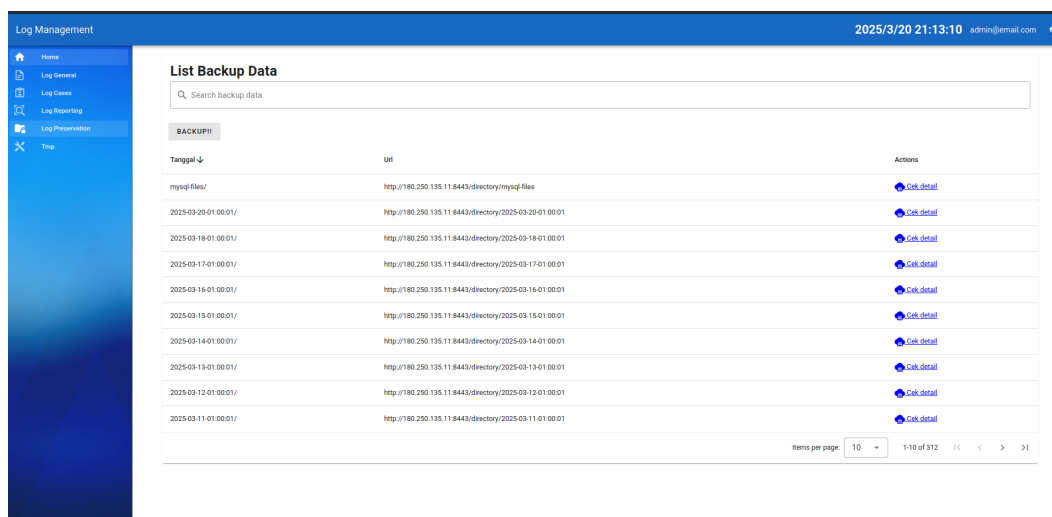


Figure 4.14: Notification

Fig 4.14 notification feature through Telegram bots is to facilitate better monitoring and response in an online examination system. Notification of the proactive analysis log results will be sent via telegram.

However, the current implementation still requires manual intervention to trigger the notification to administrators or proctors. That is, although the log analysis system flags a suspicious case, an operator must manually confirm and forward the alert. This limitation reduces the level of automation in the incident response process. Future development should consider implementing a fully automated alerting mechanism, where notifications are sent immediately upon detection of a suspicious event.

4.4.8 Log Preservation

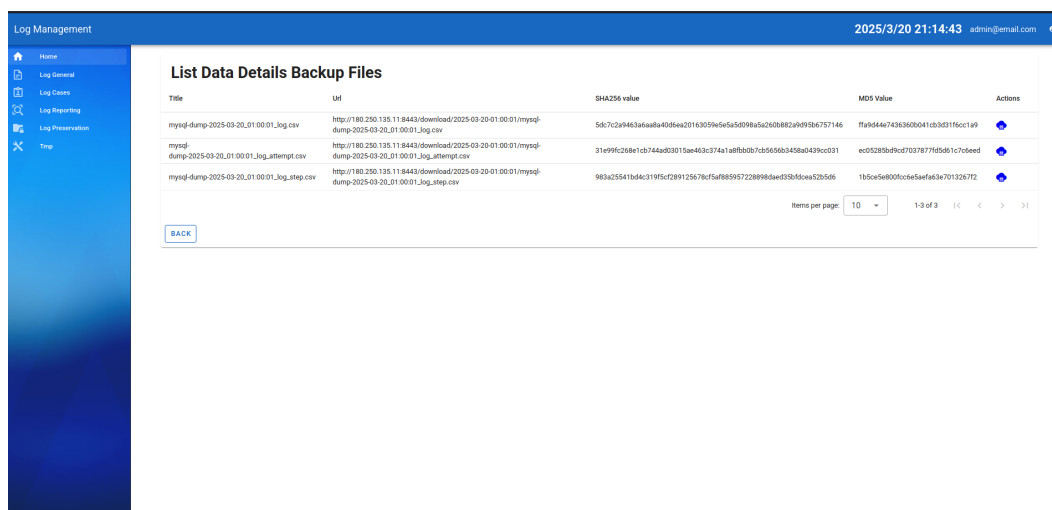


The screenshot shows the 'Log Management' interface with a sidebar menu. The main content area is titled 'List Backup Data' and contains a search bar and a table of backup data. The table has columns for 'Tanggal' (Date), 'Url', and 'Actions'. The data shows a series of backups from 2025-03-20 to 2025-03-11.

Tanggal	Url	Actions
mysql-files/	http://180.250.135.11:8443/directory/mysql-files	Click detail
2025-03-20-01:00:01/	http://180.250.135.11:8443/directory/2025-03-20-01:00:01	Click detail
2025-03-19-01:00:01/	http://180.250.135.11:8443/directory/2025-03-19-01:00:01	Click detail
2025-03-17-01:00:01/	http://180.250.135.11:8443/directory/2025-03-17-01:00:01	Click detail
2025-03-16-01:00:01/	http://180.250.135.11:8443/directory/2025-03-16-01:00:01	Click detail
2025-03-15-01:00:01/	http://180.250.135.11:8443/directory/2025-03-15-01:00:01	Click detail
2025-03-14-01:00:01/	http://180.250.135.11:8443/directory/2025-03-14-01:00:01	Click detail
2025-03-13-01:00:01/	http://180.250.135.11:8443/directory/2025-03-13-01:00:01	Click detail
2025-03-12-01:00:01/	http://180.250.135.11:8443/directory/2025-03-12-01:00:01	Click detail
2025-03-11-01:00:01/	http://180.250.135.11:8443/directory/2025-03-11-01:00:01	Click detail

Figure 4.15: Log Preservation Directory

In the figure 4.15, it is a list that shows the date and timestamp information of activities performed by proactive log collection in the previous phase. Then, within it, there are three things obtained.



The screenshot shows the 'Log Management' interface with a sidebar menu. The main content area is titled 'List Data Details Backup Files' and contains a table of backup file details. The table has columns for 'Title', 'Url', 'SHA256 value', 'MD5 Value', and 'Actions'. The data shows three backup files: 'mysql-dump-2025-03-20-01:00:01_log.csv', 'mysql-dump-2025-03-20-01:00:01_log_attempt.csv', and 'mysql-dump-2025-03-20-01:00:01_log_step.csv'.

Title	Url	SHA256 value	MD5 Value	Actions
mysql-dump-2025-03-20-01:00:01_log.csv	http://180.250.135.11:8443/download/2025-03-20-01:00:01/mysql-dump-2025-03-20-01:00:01_log.csv	5d07c2a9443b6a8a40d5ea20143059e5e5a5098a5a260b882a999b6757146	f5a9444e7436360b041cb3d31f6c1a9	Click detail
mysql-dump-2025-03-20-01:00:01_log_attempt.csv	http://180.250.135.11:8443/download/2025-03-20-01:00:01/mysql-dump-2025-03-20-01:00:01_log_attempt.csv	31e9f9c268e1cb744ad3015ae463c374a1a8fbb0b7cb5656b3458a439cc031	ec03289bdfcd7037877f65d61c7cdeed	Click detail
mysql-dump-2025-03-20-01:00:01_log_step.csv	http://180.250.135.11:8443/download/2025-03-20-01:00:01/mysql-dump-2025-03-20-01:00:01_log_step.csv	983a25541bd4c319f5cf2b912567c7fa888957228998dced35a50a52b5d6	1b50e5e800f0c6e5aef6a3e701326772	Click detail

Figure 4.16: Detail Log Preservation

The display in the figure 4.16 shows the contents of log preservation. There are three logs that can be viewed: log attempt, log attempt step, and log general. Each log also has its hash value calculated to prevent log data changes or log tampering.

Table 4.8: Example of standard log table structure

Field	Value
action	reviewed
component	mod_quiz
course_name	EPrT HE Pre-Exam
ip	182.253.124.129
log_id	2514
quiz_id	null
quiz_name	null
target	attempt
timecreated	1736946654
user_firstname	admin
user_id	4
user_lastname	admin

- action: User action (reviewed = viewing results)
- component: Related Moodle module (mod_quiz = quiz)
- course_name: Course name
- ip: User IP address (location tracking)
- log_id: Unique log ID
- target: Action target (attempt = exam attempt)
- timecreated: Unix timestamp (seconds since 1/1/1970)
- user_lastname: User identity (ID, name)
- attempt_id: 20 - Unique identifier for this quiz attempt
- course_name: EPrT HE Pre-Exam - Name of the course containing the quiz
- firstname: admin - First name of the user who took the quiz
- lastname: admin - Last name of the user who took the quiz
- next_step_time: 1736946654 - Timestamp for when the next step should occur
- question_attempt_id: 104 - ID tracking this specific question attempt
- quiz_name: Grammar Pre-Exam - Name of the quiz attempted

Table 4.9: Quiz Attempt Log Data Example

Field	Value
attempt_id	20
course_name	EPrT HE Pre-Exam
firstname	admin
lastname	admin
next_step_time	1736946654
question_attempt_id	104
quiz_name	Grammar Pre-Exam
score	1.00000
step_id	296
step_start_time	1736946642
step_state	complete
time_spent_on_question	12 (seconds)
timefinish	1736946654
timestart	1736946639
uniqueid	20
user_id	4

- score: 1.00000 - Points earned for this question (1.0)
- step_id: 296 - Identifier for this step in the attempt
- step_start_time: 1736946642 - When this step began (Unix timestamp)
- step_state: complete - Current status of this question step
- time_spent_on_question: 12 - Seconds spent answering this question
- timefinish: 1736946654 - When attempt was completed
- timestart: 1736946639 - When attempt was started
- uniqueid: 20 - Another unique identifier for this attempt
- user_id: 4 - Moodle's internal user identifier

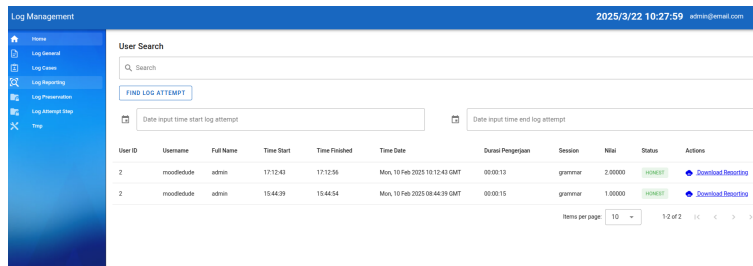
Table 4.10: Backup File Information

Field	Value
fullpath	/home/.../2024-11-27-01:00:01/ mysql-dump-2024-11-27-..._log_step.csv
md5	d41d8cd98f00b204e9800998ecf8427e
title	mysql-dump-2024-11-27-..._log_step.csv
url	http://180.xxx.xxx.xxx:8443/.../ 2024-11-27-.../mysql-dump-..._log_step.csv

The explanation of Table 4.10 is outlined as follows:

- **fullpath**: Complete server path to the backup CSV file containing MySQL log data
- **md5**: 32-character checksum for file verification (empty file indicator)
- **title**: Automated backup filename with timestamp
- **url**: Download link for retrieving the backup file

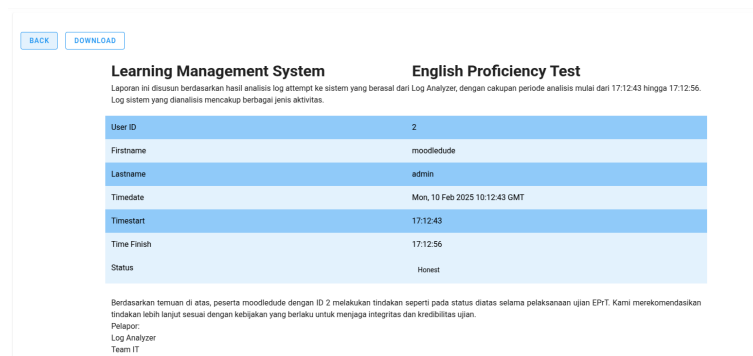
4.4.9 Reporting



The screenshot shows a 'Log Management' dashboard with a sidebar menu and a main content area. The main area displays a 'User Search' section with a search bar and a 'FIND LOG ATTEMPT' button. Below this is a table with columns: User ID, Username, Full Name, Time Start, Time Finished, Time Date, Durasi Pengujian, Session, Nilai, Status, and Actions. Two rows of data are visible for User ID 2.

User ID	Username	Full Name	Time Start	Time Finished	Time Date	Durasi Pengujian	Session	Nilai	Status	Actions
2	moodledude	admin	17:12:43	17:12:56	Mon, 10 Feb 2025 10:12:43 GMT	00:00:13	grammar	2.00000	Honest	Download Report
2	moodledude	admin	15:44:39	15:44:54	Mon, 10 Feb 2025 08:44:39 GMT	00:00:15	grammar	1.00000	Honest	Download Report

Figure 4.17: Reporting User



The screenshot shows a 'Learning Management System' report for an 'English Proficiency Test'. It includes a summary of the test results and a detailed table of user activity.

User ID	2
Firstname	moodledude
Lastname	admin
Timedate	Mon, 10 Feb 2025 10:12:43 GMT
Timestart	17:12:43
Time Finish	17:12:56
Status	Honest

Berdasarkan temuan di atas, peserta moodledude dengan ID 2 melakukan tindakan seperti pada status diatas selama pelaksanaan ujian EPT. Kami merekomendasikan tindakan lebih lanjut sesuai dengan kebijakan yang berlaku untuk menjaga integritas dan kredibilitas ujian.

Pelapor:
Log Analyzer
Team IT

Figure 4.18: Reporting User Download

The log reporting document serves as a consolidated summary of user activity during online examinations. It includes essential information such as the participant's name, timestamp of each recorded action, and a classification status indicating whether the behavior is considered suspicious. This structured report enables proctors or administrators to review potential anomalies efficiently and supports the decision-making process for further investigation or enforcement actions. The inclusion of timestamped events and flagged indicators enhances the traceability and forensic value of the evidence collected.

4.5 Phase 5

Phase 5 focuses on evaluating the implemented forensic log management framework through simulated forensic scenarios. This evaluation aims to determine whether the system is capable of

supporting proactive forensic readiness—particularly in the context of online examination environments where incidents such as impersonation, unauthorized access, or rapid submission attempts may occur.

4.5.1 Verification

The results obtained from the experiment using the log management method revealed the phases from identification to reporting logs in online exams. The results can be seen in the table below.

Table 4.11: Verifying testing framework adopted from NIST 800-92

Phase	Expected Result	Result
1. Log Identification	All log sources from Moodle and server are identified, including quiz attempts and activity logs	As Expected
2. Log Proactive Collection	Automated scripts collect logs daily from exam sessions without disrupting system performance	As Expected
3. Log Transmission	Log files transferred via rsync over SSH securely and consistently	As Expected
4. Log Storage	Log data is stored in centralized, timestamped folders with access control and retention policy	As Expected
5. Log Analyzer	Dashboard successfully displays log data with filtering, classification, and visualization features	As Expected
6. Log Proactive Analysis	Anomaly detection using machine learning identifies suspicious patterns from log data	As Expected
7. Send Notification	Telegram bot sends alert based on flagged anomalies from the dashboard to the administrator	As Expected
8. Log Preservation	Logs are stored with integrity checks (MD5 hash) to ensure tamper-evidence	As Expected
9. Log Reporting	PDF report is generated, presenting user activity and anomaly classification in structured format	As Expected

Table 4.11 can be used to verify that the framework adopted from NIST 800-92 works as expected. The verification results of the framework show that each phase has been carried out

according to its objectives and has achieved results that align with what was intended.

4.6 Phase 6

Phase 6 is the final stage in the development lifecycle, focusing on the validation of the proposed forensic log management framework by relevant domain experts. The purpose of this phase is to assess the framework's technical soundness, practical applicability, and completeness when applied in real-world online examination environments.

4.6.1 Validation

To assess the effectiveness and practicality of the developed proactive forensic log management system, a validation process was conducted involving a panel of domain experts. The purpose of this validation was to evaluate the system's alignment with digital forensic principles, its technical reliability, and its suitability for use in online examination environments.

Justification for Using Interviews: The interview method was chosen because it enables deeper exploration of expert perspectives that may not be captured through quantitative surveys. Given the complexity and domain-specific nature of the system particularly in relation to log integrity, forensic readiness, and anomaly detection semi-structured interviews allow experts to elaborate on technical insights and provide contextual evaluations. This method is also appropriate for validating design decisions in early-stage frameworks where practical deployment feedback is critical.

Main Questions Asked:

- Is the scope of log data collected sufficient for forensic investigation?
- Does the system maintain the integrity and traceability of log data effectively?
- How reliable is the anomaly detection approach based on machine learning?
- Are the notification and reporting features helpful for exam monitoring?
- Could this system be realistically deployed in a real online examination environment?

4.7 Result of framework log management



Figure 4.19: Phase log management from NIST 800-92 2006 Kent and Souppaya [13]

Fig 4.19 original from publication nist 800-92 2006.



Figure 4.20: Proactive Forensics Alharbi et al. [4]

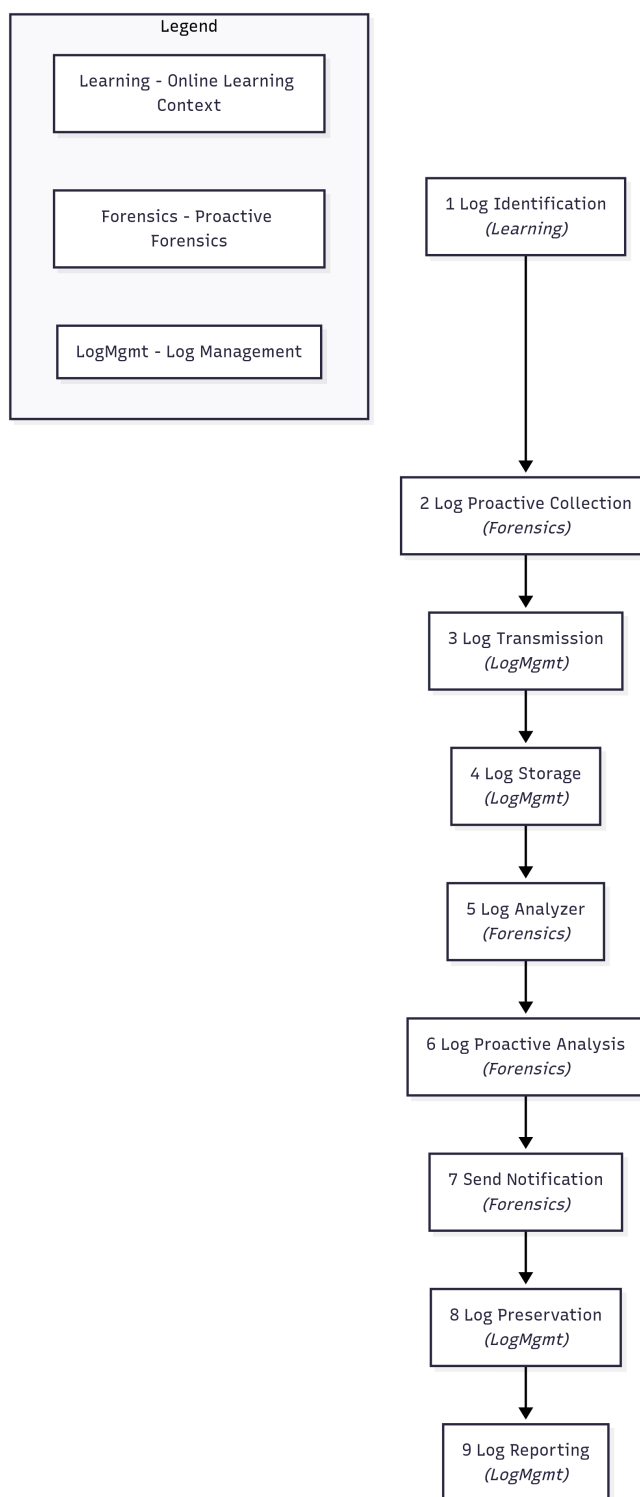


Figure 4.21: Proposed framework adapted from NIST 800-92

Figure 4.21 is the result of an adaptation based on NIST 800-92 regarding log management. The proposed framework adopted from NIST SP 800-92 (Guide to Computer Security Log Management)

provides a structured approach to managing and analyzing logs.

4.8 Summary of Findings

Table 4.12: Comparison of Proactive and Reactive Forensic Based on Log Management Aspects

Aspect	Proactive Forensic	Reactive Forensic
Log Generation	Logging is configured in advance with consistent policies to capture all relevant events, even before incidents occur.	Logging may not be fully enabled until after an incident is detected or suspected.
Log Transmission	Logs are periodically transmitted to a centralized repository as part of ongoing readiness.	Logs may be manually collected from devices post-incident; transmission is reactive and possibly delayed.
Log Storage	Logs are stored securely with defined retention periods, using structured directories and integrity-preserving mechanisms.	Logs may be scattered or incomplete; storage begins or is prioritized after an incident is identified.
Log Analysis	Analysis is conducted before incidents occur, aiming to identify anomalies and potential threats early (proactive log analysis).	Analysis is difficult if logs are incomplete or missing; lack of evidence may hinder investigation.

Table 4.12 compares proactive and reactive forensic approaches based on key aspects of log management. In proactive forensics, log generation is pre-configured with consistent policies to ensure comprehensive event capture before incidents occur, whereas reactive forensics often lack complete logging until an incident is suspected. Proactive approaches also include scheduled log transmission to a centralized repository, secure storage with structured organization and integrity measures, and early-stage log analysis to detect anomalies. In contrast, reactive methods typically rely on delayed or manual log collection, ad-hoc storage, and limited analysis capabilities due to incomplete or missing logs, which can hinder effective investigations.

Table 4.13: Log Management Processes

No	Process	Method	Challenges	Output
1	Log Identification	Identify all log sources	Legacy systems with non-standard formats (e.g., CSV, plaintext)	List of log sources
2	Log Proactive Collection	Automated daily backup scripts	Risk of database server overload during peak hours	Daily backups of quiz attempt logs
3	Log Transmission	File transfer using rsync protocol over SSH	Network latency, ensuring file consistency	Synchronized log files in log storage server
4	Log Storage	Centralized structured directory with access control	Scalability, retention policy enforcement, integrity preservation, and filtering irrelevant logs (e.g., unrelated Apache2 web server logs)	Organized, timestamped log archive
5	Log Analyzer	Custom dashboard development	Integration of multiple log formats into unified view	Log activity visualized via web dashboard
6	Log Proactive Analysis	Log anomaly detection using ML (Isolation Forest)	High computation and tuning threshold values	Preliminary anomaly classification
7	Send Notification	Telegram bot integration for alerts	Notification workflow still relies on manual confirmation	Cheating alert notification for administrators
8	Log Preservation	CSV export with MD5 hash checksum	Disk I/O load, ensuring file immutability	Verified and tamper-evident log files
9	Log Reporting	Document generation via dashboard export	Report standardization and formatting issues	PDF-based user activity reports

Table 4.13, Log management initiates with Log Identification for comprehensive source mapping, frequently encountering interoperability issues with legacy system formats. Subsequent Log Proactive Collection utilizes automated daily backup mechanisms, presenting potential database server performance impacts. Analysis is conducted through a Custom Dashboard (Log Analyzer).

The log management phase is structured in the following sequence to support proactive forensic readiness:

1. **Database Log Sources**

The process begins with capturing high-relevance log data from the database, such as quiz attempts and user session activity, which serve as primary sources for digital forensic analysis.

2. **Periodic Proactive Log Collection**

Log data is periodically extracted from the database and other sources through automated mechanisms to ensure consistent and up-to-date monitoring of user activities.

3. **Log Storage**

Collected logs are then stored in a centralized, structured, and secure directory system, with mechanisms for timestamping, access control, and integrity verification.

CHAPTER 5

Conclusion and Recommendations

5.1 Conclusion

This study demonstrates that proactive forensics is a viable approach for enhancing the integrity and auditability of online examination systems. By enabling the collection of log data prior to the occurrence of suspicious activities, proactive forensics ensures the availability and reliability of digital evidence for further analysis. The proactive collection process is central to this capability, as it facilitates continuous monitoring and automated acquisition of log data, including user interactions, quiz attempts, and system-generated records from the Moodle platform.

The implementation of a framework based on NIST Special Publication 800-92 provides structured guidance in managing logs systematically. Integrating this standard with proactive forensic techniques improves the readiness and responsiveness of digital forensic activities, especially in academic environments. This integration supports secure log acquisition, centralized storage, anomaly detection, and evidence preservation.

Furthermore, the application of machine learning, particularly anomaly detection using Isolation Forest, enhances the ability to identify potentially fraudulent behavior that might be overlooked through manual inspection. Overall, the proposed system contributes to the advancement of forensic readiness by combining structured log management with intelligent analysis mechanisms.

5.2 Recommendations

Based on the findings and conclusions of this study, several recommendations are proposed to support further development and application of proactive forensic systems:

- Future work should explore the integration of alternative log management frameworks or technologies to improve adaptability and performance in various academic settings.
- Comparative studies with other proactive forensic techniques should be conducted to evaluate effectiveness and scalability in broader deployment scenarios.
- Since this research was conducted as a controlled prototype simulation, it is recommended that subsequent implementations be tested in live academic environments to validate system robustness and practical utility.
- Additional work could also focus on enhancing the anomaly detection mechanism by experimenting with different machine learning models or incorporating more granular behavioral metrics.

These recommendations are intended to support future research and practical implementation efforts toward achieving a robust, scalable, and forensic-ready online examination system.

BIBLIOGRAPHY

- [1] N. Abd Hamid, N. H. Ab Rahman, and N. D. W. Cahyani. Enhancing learning management systems with intrusion alerts and forensic logging. *International Journal of Advanced Research in Education and Society*, 6(3):295–308, 2024.
- [2] A. Adel, A. Ahsan, and C. Davison. Ethicore: Ethical compliance and oversight framework for digital forensic readiness. *Information*, 15(6):363, 2024.
- [3] M. Al-Fayoumi and S. J. Aboud. An efficient e-exam scheme. *Int. J. Emerg. Technol. Learn.*, 12(4):153–162, 2017.
- [4] S. Alharbi, J. Weber-Jahnke, and I. Traore. The proactive and reactive digital forensics investigation process: A systematic literature review. In *Information Security and Assurance: International Conference, ISA 2011, Brno, Czech Republic, August 15-17, 2011. Proceedings*, pages 87–100. Springer, 2011.
- [5] J. Barker and M. Tan. Open-source log management tools as siem alternatives in small enterprises. *International Journal of Cyber Security and Digital Forensics*, 7(4):201–209, 2018.
- [6] V. M. Bradley. Learning management system (lms) use with online instruction. *International Journal of Technology in Education*, 4(1):68–92, 2021.
- [7] S. Davidovič and K. Guliani. Reliable cron across the planet. *ACM Queue*, 13(3), Mar. 2015.
- [8] R. Febriana, A. Luthfi, et al. Comparative study of cloud forensic investigation using adam and nist 800-86 methods in private cloud computing. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 7(5):1097–1110, 2023.
- [9] M. Garg and A. Goel. Preserving integrity in online assessment using feature engineering and machine learning. *Expert Systems with Applications*, 225:120111, 2023.
- [10] I. H. A. Halim, A. R. M. Saad, N. F. N. Ramli, and A. N. M. Rozaini. Reducing honey-pot log storage capacity consumption – cron job with perl-script approach. arXiv preprint arXiv:1911.07633, 2019. URL <https://arxiv.org/abs/1911.07633>.
- [11] G. Johansen. *Digital forensics and incident response*. Packt Publishing Ltd, 2017.
- [12] N. Kadoic and D. Oreski. Analysis of student behavior and success based on logs in moodle. 2018 41st international convention on information and communication technology, electronics and microelectronics, mipro 2018-proceedings,(pp. 654–659), 2018.
- [13] K. A. Kent and M. Souppaya. Guide to computer security log management:. 2006.

- [14] M. Kern, M. Landauer, F. Skopik, and E. Weippl. A logging maturity and decision model for the selection of intrusion detection cyber security solutions. *Computers & Security*, 141: 103844, 2024.
- [15] A. Kumar. Scalability and security issues in learning management systems. *IEEE Access*, 7: 150985–150995, 2019.
- [16] V. Lakhno, S. Adilzhanova, M. Ydyryshbayeva, A. Turgynbayeva, O. Kryvoruchko, V. Chubaievskiy, and A. Desiatko. Adaptive monitoring of companies’ information security. *International Journal of Electronics and Telecommunications*, 69(1), 2023.
- [17] D. Larchenko and O. Shevchenko. Mind map-based classification of challenges in digital forensics. *Journal of Information Security Research*, 18(2):112–120, 2025. doi: 10.1007/s10207-025-00621-x. Used as basis for visualizing evidence loss categories in digital forensics.
- [18] Q. K. Lintang, N. D. W. Cahyani, and M. A. Nugroho. Log analyzer for english proficiency test cheating detection. In *2024 12th International Conference on Information and Communication Technology (ICoICT)*, pages 115–122. IEEE, 2024.
- [19] V. Machaka and T. Balan. Investigating proactive digital forensics leveraging adversary emulation. *Applied Sciences*, 12(18):9077, 2022.
- [20] L. Nussbaum et al. Performance comparison of scp and rsync over variable latency and bandwidth. In *Proceedings of ...*, 2012.
- [21] D. Paul Joseph and J. Norman. An analysis of digital forensics in cyber security. In *First International Conference on Artificial Intelligence and Cognitive Computing: AICC 2018*, pages 701–708. Springer, 2019.
- [22] D. Putra and M. Sari. Integration of artificial intelligence in learning management systems. In *Proceedings of the National Seminar on Information Technology*, pages 98–105, 2022.
- [23] J. Ranger, N. Schmidt, and A. Wolgast. The detection of cheating on e-exams in higher education—the performance of several old and some new indicators. *Frontiers in Psychology*, 11:568825, 2020.
- [24] F. Rivera-Ortiz and L. Pasquale. Towards automated logging for forensic-ready software systems. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 157–163. IEEE, 2019.
- [25] Y. Rosmansyah, M. Ritonga, and A. Hardi. An attack-defense tree on e-exam system. *International Journal of Emerging Technologies in Learning (iJET)*, 14(23):251–260, 2019.
- [26] Y. Rosmansyah, I. Hendarto, and D. Pratama. Impersonation attack-defense tree. *International Journal of Emerging Technologies in Learning (iJET)*, 15(19):239–246, 2020.

- [27] A. Sivaprasad. Secured proactive network forensic framework. In *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, pages 695–699, 2017. doi: 10.1109/CTCEEC.2017.8455003.
- [28] L. K. Smirani and J. A. Boulahia. An algorithm based on convolutional neural networks to manage online exams via learning management system without using a webcam. *International Journal of Advanced Computer Science and Applications*, 13(3), 2022.
- [29] J. Smith and P. Brown. A comparative study of moodle and google classroom in higher education. *International Journal of Information and Education Technology*, 9(2):112–117, 2020.
- [30] StackExchange community. Rsync and scp differences. <https://serverfault.com/questions/194514/rsync-and-scp-differences>. Accessed July 2025.
- [31] StackExchange SuperUser community. Why is rsync faster than scp? <https://superuser.com/questions/193952>. Accessed July 2025.
- [32] K. Sylla, B. Babou, and S. Ouya. Secure dematerialization of assessments in digital universities through moodle, webrtc and safe exam browser (seb). *International Association for Development of the Information Society*, 2022.
- [33] H. Venter and I. Kigwana. A digital forensic readiness architecture for online examinations. *South African Computer Journal*, 30(1):1–39, 2018.
- [34] Y. Wang and X. Liu. A lightweight alerting system using telegram bot api for network anomaly detection. *International Journal of Computer Applications*, 185(12):45–51, 2023.
- [35] R. Wijaya, S. Nugroho, and L. Pratama. Mobile-based lms usage in distance learning. *Jurnal Teknologi Pendidikan*, 5(1):21–30, 2021.

Appendices

APPENDIX A

Hasil Wawancara dengan Staf IT

Wawancara ini dilakukan dengan seorang staf dari tim IT pada hari Senin, 2 Juni 2025, secara tatap muka. Tujuan dari wawancara ini adalah untuk melakukan validasi terhadap kendala ketika melakukan investigasi pencarian log, khususnya dalam aspek pengelolaan log (log management) pada sistem ujian online berbasis Moodle.

Pertanyaan 1

Bagaimana cara mengambil data dari sumber log kecurangan?

Jawaban:

1. **Metode:** Untuk pengambilan log dilakukan satu per satu. Untuk log dari aplikasi, prosesnya dilakukan melalui antarmuka Moodle dengan masuk ke bagian *quiz* dan *course*. Untuk database, digunakan referensi data dari tabel log standar (general log). Sedangkan untuk data attempt (percobaan ujian), dibutuhkan query tambahan yang menggabungkan beberapa tabel.
2. **Kesulitan:** Terdapat kendala pada jumlah log yang sangat banyak (membengkak), sehingga pengolahan log masih bersifat reaktif. Log retention yang diterapkan bervariasi: log quiz attempt hanya disimpan selama 3 hari, log aplikasi (termasuk database) disimpan selama 2 tahun, sedangkan log dari layanan Cloudflare hanya disimpan selama 7 hari. Karena quiz sudah terintegrasi dengan sistem iGracias, log retention hanya 1 hari.
3. **Output:** Harapan dari proses ini adalah agar penggunaan log dapat dilakukan secara terentral untuk memudahkan analisis.

Pertanyaan 2

Bagaimana mekanisme penyimpanan log (on-premise/cloud, terenkripsi/tidak)?

Jawaban:

1. **Metode:** Mekanisme penyimpanan log dilakukan pada cloud menggunakan VM storage untuk data log seperti Apache dan layanan web. Sedangkan untuk log dari database disimpan pada VM database yang juga berada di cloud.
2. **Kesulitan:** Tantangan yang dihadapi berkaitan dengan manajemen storage, terutama pada sistem multi-instance. Akses log antar instance dilakukan melalui NFS (Network File System). Proses sinkronisasi log dilakukan setiap menit menggunakan `rsync`, yang menyebabkan

pembengkakan data. Selain itu, duplikasi log antar sistem menambah beban penyimpanan dan menyebabkan ukuran storage membengkak.

3. **Output:** Harapannya adalah log dapat dikelola secara lebih terpusat (centralized) dan efisien dalam penggunaan storage. Namun, penggunaan kompresi (zipping) untuk efisiensi ruang penyimpanan berdampak pada penggunaan CPU yang tinggi, karena semakin tinggi kompresi, semakin besar beban pemrosesan (CPU usage).