

Large Language Models in a Local Environment for Generative AI-Based Automated Event Log Processing in Process Mining Techniques

Poohridate Arpasat * and Wichian Premchaiswadi 

Graduate School of Information Technology, Siam University, Bangkok, Thailand

Email: poohridate@siam.edu (P.A.); wichian@siam.edu (W.P.)

*Corresponding author

Abstract—Process mining is vital for enhancing the operational efficiency and supporting strategic decision-making of organizations because it identifies bottlenecks and optimizes workflows. However, its analytical success depends fundamentally on the quality of event logs, and preparing these logs is a highly complex task. This complexity stems from diverse data storage formats across organizational systems, including varied log structures and inconsistent timestamps. Thus, analysts utilize public generative Artificial Intelligence (AI) services for data preparation, which demonstrates functional capabilities within usage limitations. However, the transmission of sensitive data to external servers in public generative AI systems poses data leakage risks. Moreover, manual intervention required in compiling the AI model-exported data is prone to errors and time consuming. Token limit violations in such models also cause service interruptions. To address these challenges, a local generative AI system was developed for automated event log transformation utilizing Ollama Llama 3.1. The system employed Python with the Streamlit framework, Pandas library, and Ollama integration. It thus implemented rule-based format detection algorithms, AI-assisted semantic column mapping, and a data transformation engine that managed complex conversions, including regular expression algorithms for timestamp standardization. This system was tested on real data obtained from a hospital in Thailand (128,917 events from 10,217 cases), during which it processed 4,194 events per second. It transformed complex data formats into standard event logs, which were then imported into a process mining software, Disco. Results indicated that this system addressed data security concerns by establishing automated processes, enabling organizations to leverage the benefits of AI in preparing data for process mining without data leakage risks.

Keywords—process mining, event log preprocessing, large language model, Ollama Framework, Artificial Intelligence (AI) local environment, automated data transformation

I. INTRODUCTION

Process mining is used to analyze real data from information systems to visualize work patterns, identify

bottlenecks, and propose data-driven improvement approaches for businesses [1]. This approach has helped to reduce operational costs and increase process efficiency across organizations [2]. However, the success of analysis depends on the quality of event log data. For effective process mining, these data must be structured into a specific format that includes case ID, activity, timestamp, and resource [3].

Raw data from organizational systems rarely conform to the process mining standards, posing several key challenges. These include transforming complex data structures, standardizing multilingual and inconsistent terminologies, and parsing varied timestamp formats into a unified format. Consequently, analysts spend 60–90% of their project time on data preparation [2]. In addition, manual data transformation is not only time-consuming but also prone to errors, posing challenges for novice analysts in particular [3].

Many analysts have therefore adopted online Artificial Intelligence (AI) tools such as ChatGPT or Claude for data organization and transformation. While these tools are convenient to use, sharing organizational data with external services poses serious security and privacy risks [4]. Sensitive data such as customer information, business processes, and financial data may be leaked or misused by such services. Moreover, the terms of service of many AI platforms permit the storage and utilization of user data, violating personal data protection regulations [4].

To address these issues, a local Large Language Model (LLM)-based system that uses the Ollama framework for automated event log transformation was proposed herein. This system can overcome the challenges of manual workflows by automatically detecting data patterns, transforming data into formats compatible with process mining, and efficiently handling multilingual data problems. As such, data are processed within the organization without using any external tools while also exploiting AI capabilities. In doing so, the proposed system reduces the aforementioned barriers in process mining while strengthening data security. Its effectiveness was validated based on key metrics such as processing speed to address the challenge of significant

time consumption in manual data preparation, data integrity, and practical usability.

II. BACKGROUND

Process mining and AI have garnered considerable research attention in the past decade, particularly from organizations focusing to improve the business process efficiency via in-depth data analysis. This literature review surveys studies that have used AI and LLMs for event log data preparation for process mining, with minimal processing times and high data security.

A. Process Mining

The field of process mining was developed by Wil van der Aalst and his team. It combines data mining and Business Process Management (BPM) to discover, analyze, and improve actual business processes using event log data that contain records of operational activities in information systems [3]. Process mining mainly involves (1) process discovery, which involves creating process models using event log data, (2) conformance checking, in which differences between the model and actual process behaviors are compared, and (3) process enhancement, which aims to improve the efficiency of models based on the findings of the last two processes. The corresponding framework is shown in Fig. 1.

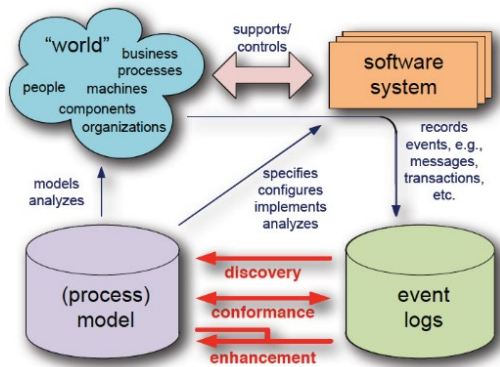


Fig. 1. Process mining framework [3].

A key feature of process mining is that it can provide useful in-depth information about organizational processes, and the quality of event log data is crucial for ensuring high accuracy of process mining [2]. Organizations implementing process mining can identify bottlenecks in processes, reduce operational time, and enhance efficiency. Various process discovery algorithms have been developed to address challenges with real-world data. For instance, the Fuzzy Miner algorithm is used in many commercial tools, including Disco. Unlike earlier algorithms that produce overly complex and hard-to-read diagrams (often called “spaghetti models”) from real-world data, Fuzzy Miner is designed to simplify process maps. It works by abstracting less-significant details and focusing on the most frequent and important activities and pathways. As a result, Fuzzy Miner offers a balanced and understandable view of the process, making it highly effective for analyzing complex and highly variable datasets such as those found in fields, including healthcare [5].

B. Event Logs

High-quality event logs for process mining must comprise four main components. (1) Case ID, which is a unique identifier of each process instance, (2) Activity, which represents the activities performance, (3) Timestamp, which records when events occur, and (4) Resource, which shows the person or system that performs the activity. This standard format is shown in Fig. 2. However, data obtained from real organizational systems do not conform to this standard format.

Marin-Castro and Tello-Leal [1] classified event log data preparation techniques into six main groups: enriching (data enrichment), integration (data integration), filtering (data filtering), transformation (data transformation), reduction (data reduction), and abstraction (data abstraction). Each technique has different advantages and limitations, requiring specific expertise in selection and use. Martin *et al.* [6] developed the DaQAPO tool for the systematic assessment of event log quality; this tool comprehensively examines temporal ordering and timestamp consistency.

	Case ID	Timestamp	Activity	Attributes			
	CaseID	Timestamp	Medium	Activity	Service Line	Urgency	
Instances	1 case9700	20.8.09 11:46	Phone	Registered	1st line	0	Events
	2 case9700	20.8.09 11:50	Phone	Completed	1st line	0	
	3 case9701	23.9.09 12:23	Phone	Registered	1st line	0	
	4 case9701	23.9.09 12:27	Phone	Completed	1st line	0	
	5 case9705	20.10.09 14:21	Phone	Registered	Specialist	2	
	6 case9705	20.10.09 16:48	Phone	At specialist	Specialist	2	
	7 case9705	19.11.09 10:31	Phone	In progress	Specialist	2	
	8 case9705	19.11.09 10:32	Phone	Completed	Specialist	2	
	9 case3939	15.10.09 11:48	Mail	Registered	Specialist	2	
	10 case3939	15.10.09 11:48	Mail	Offered	Specialist	2	
	11 case3939	20.10.09 17:18	Mail	In progress	Specialist	2	
	12 case3939	20.10.09 17:19	Mail	At specialist	Specialist	2	
	13 case3939	21.10.09 14:49	Mail	In progress	Specialist	2	
	14 case3939	21.10.09 14:49	Mail	In progress	Specialist	2	
	15 case3939	28.10.09 10:17	Mail	In progress	Specialist	2	
	16 case3939	28.10.09 10:18	Mail	Completed	Specialist	2	
	17 case9704	20.10.09 14:19	Mail	Registered	1st line	0	
	18 case9704	20.10.09 14:24	Mail	Completed	1st line	0	
	19 case9703	20.10.09 14:40	Phone	Registered	1st line	0	
	20 case9703	20.10.09 14:58	Phone	Completed	1st line	0	
	21 case9702	24.8.09 12:24	Mail	Registered	2nd line	2	
	22 case9702	24.8.09 12:30	Mail	Offered	2nd line	2	
	23						

Fig. 2. Event recording format [3].

A significant challenge with current data preparation methods is the excessive time and resource consumption. Analysts spend 60–90% of project time on data preparation [1, 2]. In the healthcare sector, integrating data from disparate systems such as patient administration, laboratory information systems, and billing often requires extensive manual scripting. This process delays the generation of analytical insights from data for weeks. This ultimately delays the project timelines and creates a significant barrier that prevents small and medium organizations from effectively implementing process mining. Moreover, manual data preparation is prone to errors that may impact the accuracy of analysis results. Fahland [7] proposed a framework for transforming raw data from organizational databases into structured event logs by providing a mathematical foundation for complex data transformation processes.

C. Artificial Intelligence and Large Language Models

The use of AI and LLMs in data processing has garnered considerable attention in the past few years, particularly after the success of GPT and other LLMs. LLMs can efficiently understand and manage unstructured data as well as transform them into different formats. Berti and Aalst [8] recently reported on the effectiveness of GPT-4 in managing object-centric process mining data and proposed effective prompting strategies. In addition, deep learning techniques such as autoencoders have been employed for improving event log quality by detecting anomalies and replacing missing data.

However, cloud-based AI services used for organizational data pose significant security risks. Parast *et al.* [9] found security concerns as the primary barrier for many organizations. Major cloud providers offer privacy-preserving configurations such as private clouds or virtual private clouds and encrypted endpoints to mitigate these risks. However, these solutions are expensive and have configuration complexities. Many organizations, particularly in sectors such as finance and healthcare, have strict regulations that limit or outrightly prohibit the transfer of sensitive data off-premises. In such cases, local LLM deployment is a superior alternative that offers security by design and ensures full regulatory compliance.

In their empirical studies, Ghosh and Grolinger [10] found that edge computing exhibited better performance than cloud-based solutions in some cases while maintaining data security. Local LLMs such as the Ollama framework help organizations to utilize AI capabilities in natural language processing and data transformation without exposing important data to external parties.

Using public AI services for processing large event log data poses critical limitations regarding token restrictions that impact their real-time usage. GPT-4 has a context window limitation of 128,000 tokens, whereas Claude-3 supports up to 200,000 tokens. However, when processing event log data with tens of thousands or

hundreds of thousands of records, these services process data by dividing large datasets into small data subsets.

This process results in loss of context between datasets, and AI models cannot maintain consistency in translating terms or categorizing activities. Moreover, processing large datasets is costly due to token usage-based pricing. The cost of processing 100,000 records using GPT-4 can be as high as 50–80 \$ per session, making this model unsuitable for regular organizational use.

D. Research Gap and Contributions

Several important research gaps have been identified, including those listed below:

- Although event log data preparation has been extensively studied, most studies have focused on developing algorithms or specific techniques without considering their usability for users without technical expertise. Data transformation from a case-centric format (commonly used in organizational databases) to an event-centric format requires complex manual operations.
- Although online AI services are widely used for data preparation, existing studies often overlook data security issues. The development of automated format detection and transformation systems that can identify diverse data formats and automatically transform them to eXtensible Event Stream (XES) standards remains an inadequately addressed need.
- Studies have not proposed an integrated approach that can handle multiple problems simultaneously such as complexity of data formats and security as well as ease of use, particularly with the use of local LLMs for processing sensitive data in organizations.

Although previous studies [8, 11, 12] have successfully demonstrated the capability of LLMs, their reliance on external Application Programming Interfaces (APIs) introduces these critical barriers: significant security and privacy risks [4], unpredictable operational costs linked to token limits, and performance bottlenecks due to network latency.

To address these barriers, a local LLM-based system has been developed. This system ensures complete data sovereignty and security, inherently complying with privacy regulations by processing all data on-premises. It also eliminates unpredictable operational costs and token-based restrictions, enabling scalable analysis without the financial and technical barriers of cloud services. This study aims to demonstrate that for core process mining tasks, the benefits of a secure, self-contained local deployment considerably outweigh the limitations of cloud-based alternatives. The proposed system adds significant practical value, enabling its enterprise adoption.

III. RELATED WORK

LLMs and generative AI are being extensively used in BPM and process mining to reduce manual efforts and

democratize process analysis. The foundational work by Estrada-Torres *et al.* [11] revealed the fundamental capability of LLMs to effectively understand and manage artifact-centric process mining, including event logs as well as procedural and declarative process models. They used natural language prompts to query process models, effectively translating human questions into formal analysis tasks. Results showed that LLMs could successfully identify bottlenecks, check for compliance, and even suggest process improvements directly from model descriptions. The primary advantage of their approach was the democratization of process analysis, allowing stakeholders without technical expertise in process mining tools to gain insights from data. However, their foundational work had limitations such as the potential for factual inaccuracies or “hallucinations” by the LLM and a lack of scalability for highly complex, enterprise-scale process models that require formal verification methods.

Building on this study, Berti and van der Aalst [8] focused on a more complex domain: Object-Centric Process Mining (OCPM). They confirmed that state-of-the-art LLMs, such as GPT-4, could effectively manage and analyze object-centric data. They primarily developed sophisticated prompting strategies specifically designed for OCPM tasks. These strategies structured the input to include information about objects and their relationships and lifecycles, guiding the LLM to perform more accurate analyses. Results revealed that these tailored prompts enabled LLMs to correctly answer complex queries about interactions between different objects (e.g., orders, packages, and invoices) within a process. In addition, simplified OCPM analysis made it more accessible to business users who may struggle with the conceptual complexity of multiobject processes. However, their approach had some limitations: its effectiveness significantly depended on the quality and specificity of prompts and it relied on powerful and often cloud-based models, inheriting their associated issues.

Sani *et al.* [12] addressed one of the most significant bottlenecks in process mining: the creation of a high-quality event log. They proposed a framework that employed Generative AI to automatically abstract business-level events from fine-grained, low-level data such as those generated by robotic process automation tools. Their method leveraged the semantic understanding of LLMs to cluster raw, low-level actions (e.g., “click button” and “copy field”) and generate meaningful, human-readable labels for these clusters (e.g., “approve purchase order”). Results revealed that their framework successfully generated an automated business-level event log with comparable quality as that of a manually created log, thereby saving significant time and effort. The main advantage was the automation of a traditionally laborious and error-prone phase, thereby enabling the analysis of processes that were previously too granular to be practical. However, a significant limitation of the framework was that the output quality depended on the ability of the LLM to infer the correct business context. Ambiguous data can lead to incorrect event labels,

potentially requiring a human-in-the-loop for validation.

Despite these promising advancements, the predominant reliance of their framework on cloud-based LLM services introduced critical barriers for its enterprise adoption. This method of sending data to external APIs carries significant security and privacy risks, including the potential leakage of sensitive business data and personally identifiable information [4]. When organizations send event logs to third-party servers, they lose direct control and create compliance challenges with regulations such as the general data protection regulation. Although major cloud service providers offer privacy-preserving configurations, they often come with increased cost and complexity [9]. Such cloud services offer access to powerful, state-of-the-art models without the need for managing infrastructure. However, they pose severe limitations in enterprise contexts: high and unpredictable operational costs tied to token usage, performance issues such as network latency and API rate limiting, and most importantly, the security risks of data exposure. These factors render public cloud solutions unsuitable for many organizations.

The confluence of the immense potential of LLMs and the severe limitations of cloud-based deployment highlights a distinct research gap: a pressing need for solutions that can harness the power of Generative AI for process mining within a secure, self-contained environment. This identified gap aligns with findings from the edge computing study of Ghosh and Grolinger [10], who showed that local data processing can offer superior performance and security. They addressed this gap by proposing and evaluating a system based on a local LLM. They ran an open-source LLM on the on-premises hardware, ensuring that the data remained within the secure network of organizations. The anticipated advantages of this system are manifold: complete data sovereignty and security, inherent compliance with privacy regulations, predictable costs (capital expenditure on hardware vs. variable operational costs), and elimination of external dependencies and network latency. However, its primary limitations are the requirement of significant in-house computational resources (e.g., GPUs) as well as technical expertise for model deployment and maintenance. Furthermore, local models may not match the raw performance of large proprietary LLMs, potentially limiting their ability to effectively handle complex tasks. We aim to demonstrate that for many core process mining tasks, the benefits of local deployment considerably outweigh these limitations.

IV. METHODOLOGY

The proposed local LLM-based system for event log data preparation can solve problems related to excessive time consumption in data preparation, security risks from using online AI services, and process mining barriers for novice analysts. The overall system workflow is shown in Fig. 3. The process begins with data ingestion, followed by AI-assisted semantic analysis for column mapping. Then, data transformation is performed, which includes

timestamp standardization and data restructuring. The process concludes with the generation of a standardized event log file.

The performance and capabilities of the system were validated on a consumer-grade laptop (HP 15s-fq2579TU), equipped with an Intel® Core™ i7-1165G7 processor, 16 GB of DDR4 RAM, and integrated Intel® Iris® Xe Graphics. Although this setup confirms the feasibility of the system on accessible, nonspecialized hardware, it introduces key operational constraints. The absence of a dedicated Graphics Processing Unit (GPU) means that the LLM inference is CPU-bound, which limits the potential throughput. Furthermore, the 16 GB memory capacity necessitates the use of a quantized version of the 7B model (e.g., 4-bit) for effective system operation. This hardware configuration, therefore, limits the scalability of the system for use with large models

(e.g., 70B) and represents a clear trade-off between broad accessibility and peak computational performance.

Fig. 3 shows the workflow, showing sequential data transformation. The workflow begins when the user uploads a raw data file (Raw CSV File) via the user interface. The semantic analysis component sends the schema and a data sample to the Ollama framework, allowing the LLM to recommend the most suitable column mapping. After the mapping is received, the data processing core takes over. First, a format detection rule-based algorithm is employed to standardize various timestamp formats while leveraging the semantic mapping output from the LLM to restructure the data, e.g., by transforming it from a case-centric to an event-centric format. The entire process operates within a secure air-gapped environment and concludes with the generation of a standardized output file ready for further analysis.

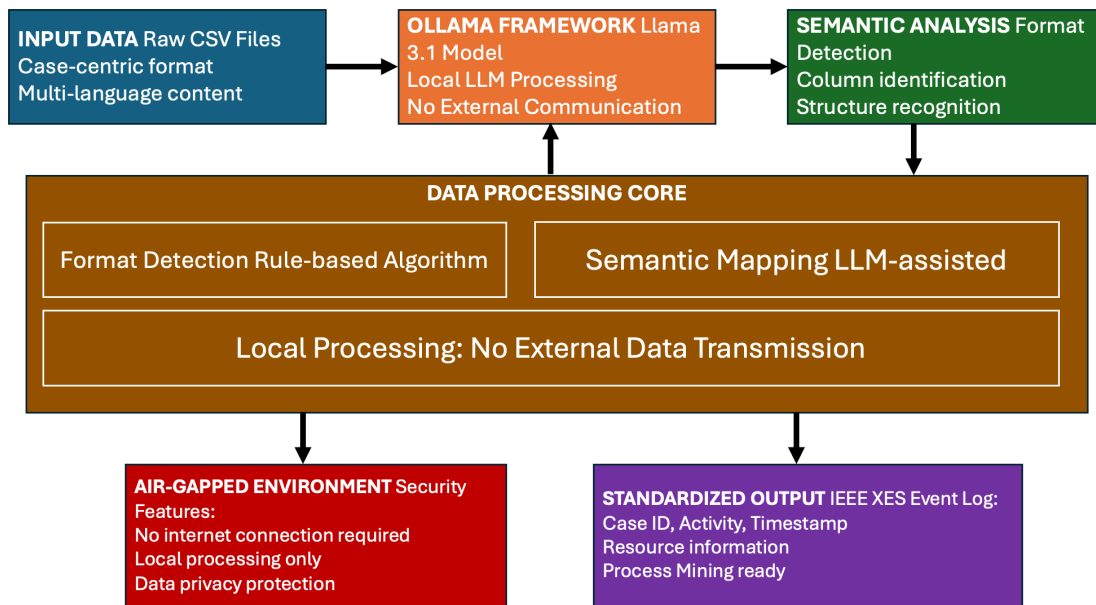


Fig. 3. Architectural overview of the local data transformation pipeline. Step-by-step process using these components.

A. Event Recording Standard Analysis

The proposed system solves specific problems involved in data preparation for process mining. The research process begins by studying and analyzing problems that analysts encounter during real-time data preparation: excessive time consumption in transforming data from various formats to event log standards and security risks from using online AI tools. Based on the event log standards for process mining following IEEE XES requirements, the data must comprise four main components (Table I).

Case ID: a unique identifier of the process instance or case.

Activity: an activity name or a step in the process.

Timestamp: date and time of activity (format: YYYY-MM-DD HH: MM: SS).

Resource: a person or a system that performs activity (optional).

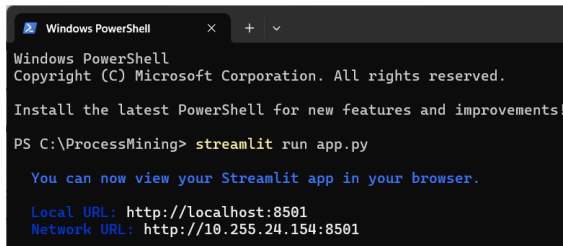
TABLE I. EXAMPLE OF CORRECT EVENT LOGS ACCORDING TO STANDARDS

Case ID	Activity	Timestamp	Resource
CASE_001	Submit Request	2024-01-15 09:00:00	John Smith
CASE_001	Review Request	2024-01-15 10:30:00	Mary Johnson
CASE_001	Approve Request	2024-01-15 14:15:00	David Brown
CASE_002	Submit Request	2024-01-15 09:15:00	Sarah Wilson
CASE_002	Review Request	2024-01-15 11:00:00	Mary Johnson

B. Developing Python Connection with Large Language Model (LLM): AI Prototype Development

The proposed system contains three main components, namely user interface layer, processing engine, and local LLM integration, programmed using the Python language.

The user interface uses the Streamlit framework to create a web-based application that can be accessed via a web browser in the local network system without internet connection. It can thus be deployed on various operating systems. Fig. 4 shows the launching of application from the command line



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements!

PS C:\ProcessMining> streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://10.255.24.154:8501
```

Fig. 4. Run Streamlit.

The processing engine—the core of the system—comprises a format detection module, data transformation engine, and quality assurance module. The format detection module uses rule-based algorithms for analyzing data structure and automatically classifying data format by examining the characteristics of columns, data types, naming patterns, and relationships between various fields. The system uses a rule-based approach for detecting data formats by analyzing the specific characteristics of data across three main dimensions: data structure, naming patterns, and field relationships.

The prototype was developed using Python with Streamlit for web interface, Pandas for data manipulation, Ollama client for LLM integration, and built-in libraries for CSV processing, as shown in the code snippet in Fig. 5. The system supports uploading CSV files via a web interface, displays data preview, allows the AI model to analyze and suggest column mapping, and exports results in a standard event log format.

```
import streamlit as st
import pandas as pd
import re
import time
import json
import hashlib
from datetime import datetime
from io import StringIO
import requests

# Process Mining Standard Schema Definition
PROCESS_MINING_SCHEMA = {
    "format": "CaseID,Activity,Timestamp,Resource",
    "example": "CASE_001,Submit_Request,2024-01-15 09:00:00,John_Smith",
    "description": ""
}

Standard Process Mining Event Log Schema:
- CaseID: Unique process instance identifier
- Activity: Business process activity name
- Timestamp: Event occurrence time (YYYY-MM-DD HH:MM:SS)
- Resource: Entity performing the activity

```

Fig. 5. Python code.

The prototype was deployed on a local web server that can be accessed from devices in the same network. It does not require internet connection after setup, making it suitable for use in environments with security limitations. The system has built-in validation for checking data quality, error handling for incomplete data, and progress tracking for processing large datasets.

The local LLM integration uses the Ollama framework to standardize complex, multilingual data via a two-stage,

data-driven process. For multilingual text fields, it employs language-agnostic sentence embeddings to group semantically equivalent activity names into clusters. It performs frequency analysis within each cluster to identify the most-common variant, which is then designated as the canonical standard. Finally, it prompts the LLM to map all low-frequency variants to this canonical form. For instance, in a cluster where “Patient Registration” is the canonical standard, a variant such as “Regis” will be standardized accordingly. For structural data such as timestamps, it uses a hybrid rule-AI approach: it applies regular expressions for common formats and leverages the LLM for exceptions, ensuring all timestamps are converted to the ISO 8601 standard. This dual capability resolves the critical challenge of low data quality in process mining.

The data transformation engine transforms data via four main steps. First, the data format is detected using the format detector module, which analyzes structure and data types using algorithm groups. Then, the transformation rule engine selects and applies appropriate transformation rules such as restructuring data into a standard event log format and standardizing timestamps. The system then conducts automatic column matching via a Natural Language Processing (NLP)-based column header analysis, analyzes sample content to identify patterns, and uses the LLM to understand the context before calculating confidence scores based on semantic similarity, pattern matching, and context relevance. For example, the system sends the column header (Patient_ID) and sample data to the LLM and instructs it to respond with a JSON file. This file automatically maps that column to the standard Case ID role based on the aforementioned metrics. For data transformation from a case-centric format to an event-centric format, the system identifies field types via user inputs or automatic assessment. It then creates event records via data pivoting in addition to case-activity-timestamp triplets. The system transforms event-centric format data via column mapping, wherein the user shares confirmation after the AI model analyzes column headers and suggests appropriate matching.

The system interface facilitates data transformation with a step-by-step wizard for guiding users. It begins with file upload, displays (Fig. 6), followed by visual data preview for users to examine the file. It then performs AI-assisted column mapping and displays the confidence scores, providing real-time feedback throughout processing. When errors or high uncertainties occur, the system switches from AI processing to rule-based processing (graceful degradation) and displays an interface for manual confirmation from the users. The system creates an automatic data backup before data transformation begins, conducts transactional processing that can be reversed, and uses incremental processing for large datasets. It then tracks real-time data quality, detects anomalies, and creates automatic quality reports before exporting the data in the standard XES format that is ready for immediate use with various process mining tools.

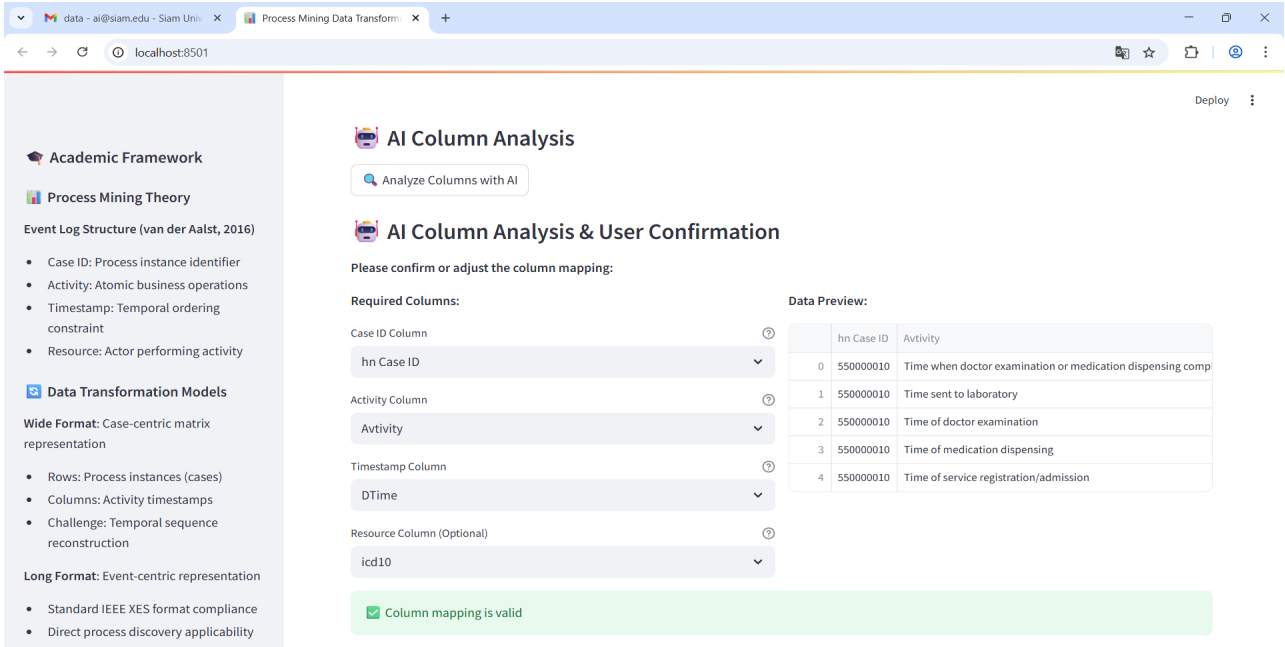


Fig. 6. Identification of field types.

C. Simulation Data Testing

Herein, two main types of datasets were used: sample data for demonstration and real-world data for performance testing.

The sample data contained two datasets. The first dataset contained repair process data obtained from a maintenance system in the case-centric format, with 49 rows and 8 columns covering case_id, activity_name, date, time, resource, cost, status, and department (Fig. 7).

The second dataset contained OutPatient Department

(OPD) process data, totaling 25 records with specific characteristics of multilingual combinations (Thai, English, and Chinese) and inconsistent timestamp formats (Fig. 8). Each record contained Patient_ID data and timestamps of various activities such as Registration, Vital_Signs_Check, Doctor_Consultation, Additional_Tests_Ordered, Receive_Test_Results, Medication_Dispensing and Payment along with Attending_Doctor, Attending_Nurse, and Dispensing_Pharmacist data.

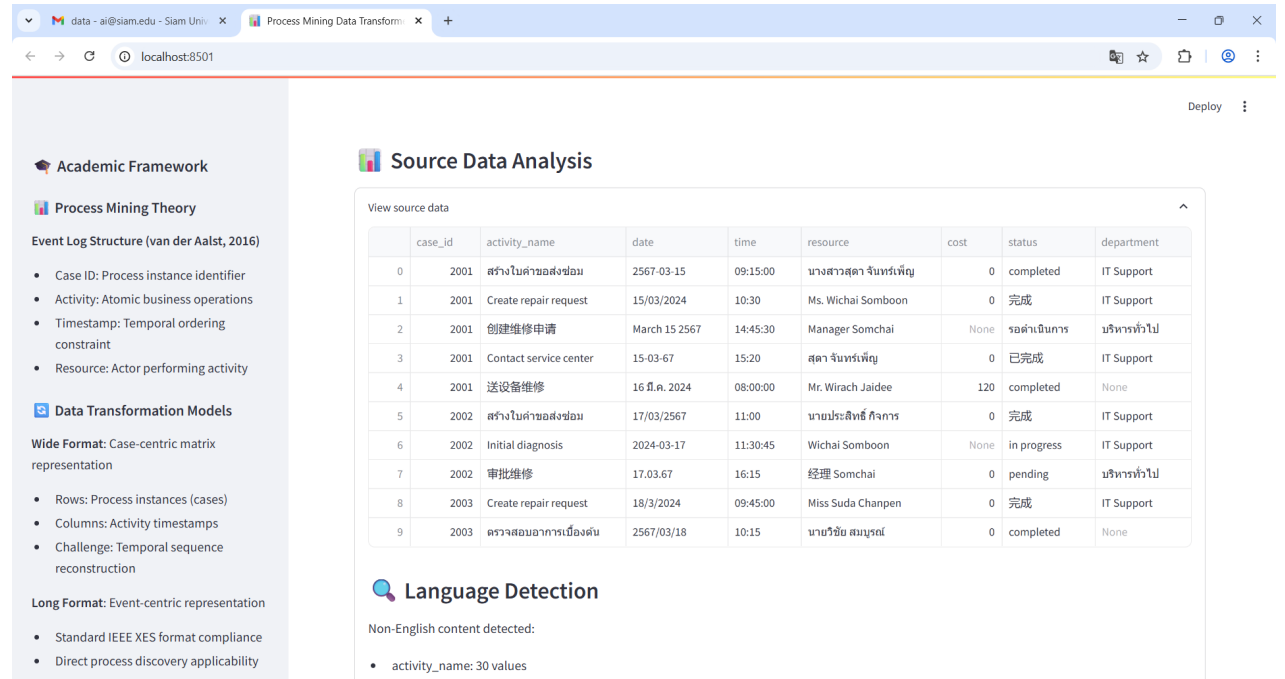


Fig. 7. Case-centric format data structure.

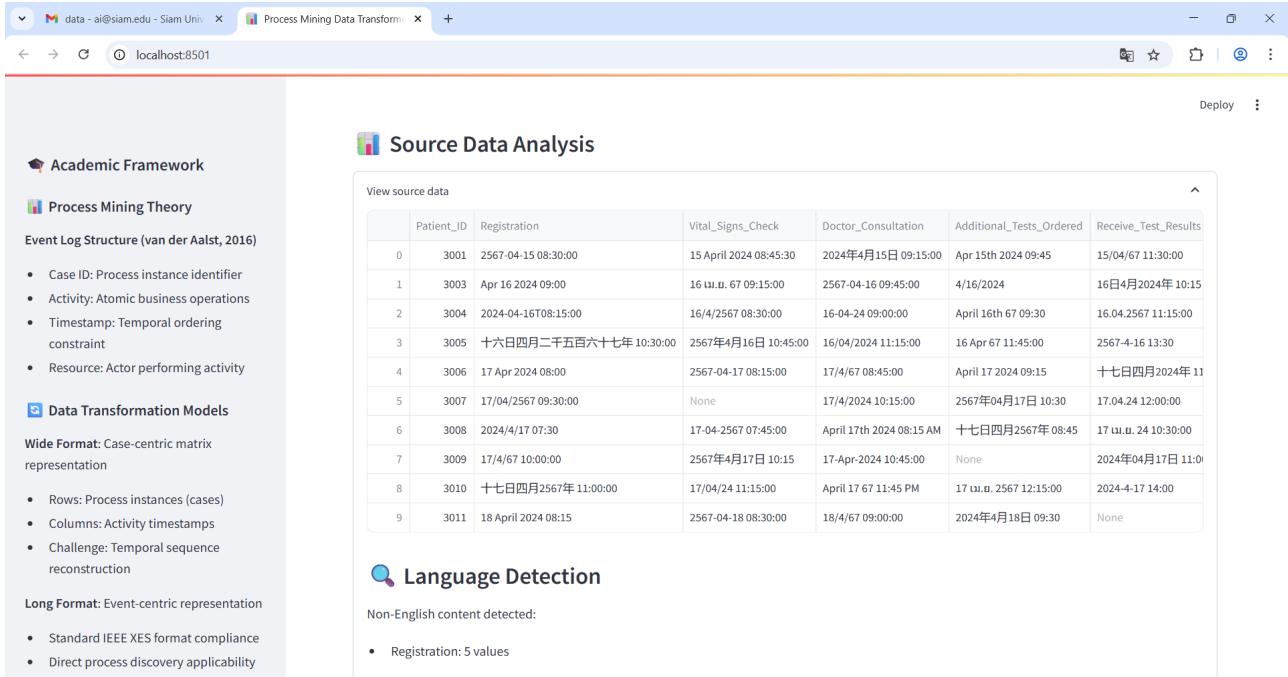


Fig. 8. Wide timestamp data format.

D. Evaluation Criteria

The performance of the developed system was evaluated based on several key qualitative criteria.

(1) Processing completion: the system must successfully process the entire test dataset comprising 128,917 events without errors to demonstrate its efficiency and stability.

(2) Transformation correctness and tool compatibility: the system performance will be assessed by verifying whether the resulting event log conforms to the standard structure (case ID, activity, and timestamp) and can be fully imported into the Disco software.

(3) Process model coherence: the process map generated in Disco must reflect a logical and realistic patient care process, confirming that the transformation by the system has genuinely preserved the integrity of the process.

V. RESULTS

The system performance was tested to evaluate its effectiveness in addressing the key challenges of manual data preparation. To this end, real data from a hospital in Thailand was obtained to measure the ability of the system to process large-scale, complex data and its compatibility with standard process mining tools.

A. System Performance Testing

The system was tested using real data obtained from outpatient care processes of a hospital in Thailand. The

dataset comprised 128,917 events from 10,217 cases covering 14 main activities. The dataset was complex because it contained information in Thai and English as well as different timestamp formats.

The automated processing of the entire dataset was completed in 30.74 s, with a processing rate of 4,194 events per second. Fig. 9 shows summary of results output on the system interface. Although the performance of this system was not directly compared with those of commercial or AI-based tools, its processing rate of 4,194 events per second on consumer-grade hardware is highly significant. Thus, the system could transform such a complex and large dataset in merely 30 s, which would otherwise require hours or even days during manual processing. The primary contribution of the system lies not only in maximizing the processing speed but also in its ability to uniquely combine automation, flexibility in handling complex formats, and the inherent data security of a fully local environment. This result is in stark contrast to traditional manual methods that take up 60–90% of a project timeline, often translating to days or even weeks for processing a dataset of such a large size and complexity [1, 2]. Compared with using public AI services, the local system eliminates issues such as token limits, the need for data chunking, and, critically, the security risks associated with sensitive data transmission. In summary, this performance demonstrates a significant improvement in efficiency and security over existing common practices.

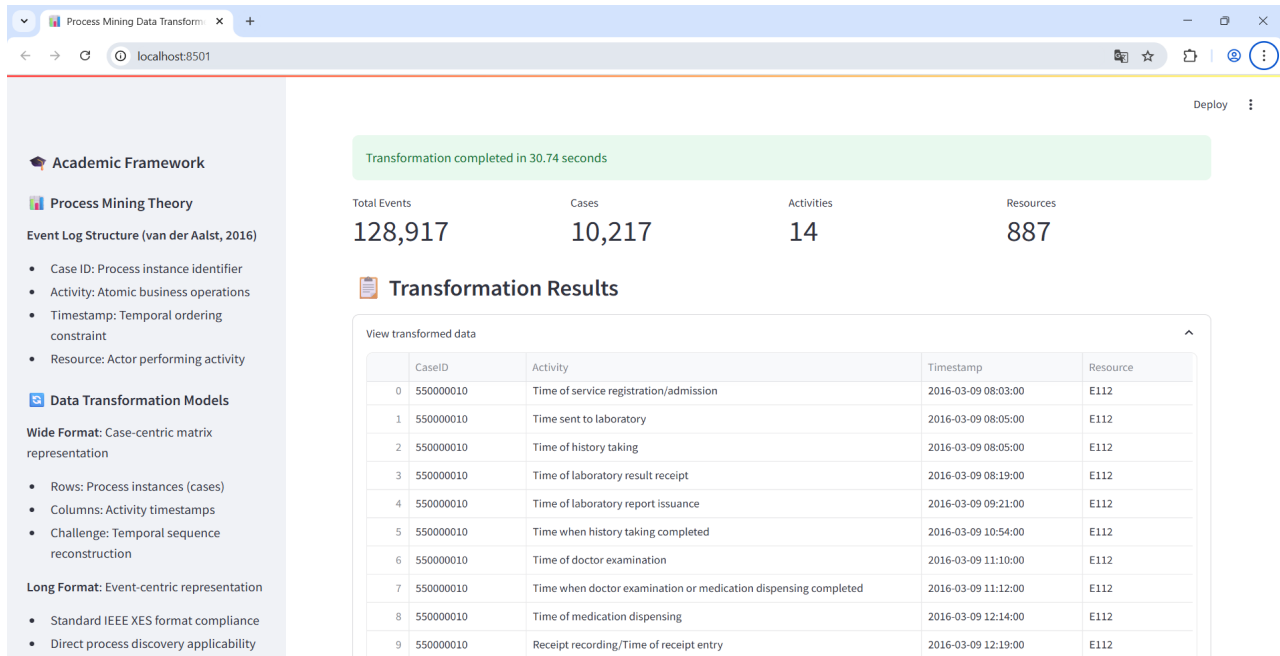


Fig. 9. Statistics of experimental results.

B. Data Transformation Results

The developed system completely transformed data from its original format to the standard event log. It automatically detected data patterns and performed transformations using highly accurate AI column mapping. The resulting event log contained complete elements according to the IEEE XES standards: case ID, activity, timestamp, and resource. A sample of the transformed data is shown in Fig. 10.

	A	B	C	D
1	CaseID	Activity	Timestamp	Resource
2	550000010	Time of service registration/admission	9/3/2016 08:03	E112
3	550000010	Time sent to laboratory	9/3/2016 08:05	E112
4	550000010	Time of history taking	9/3/2016 08:05	E112
5	550000010	Time of laboratory result receipt	9/3/2016 08:19	E112
6	550000010	Time of laboratory report issuance	9/3/2016 09:21	E112
7	550000010	Time when history taking completed	9/3/2016 10:54	E112
8	550000010	Time of doctor examination	9/3/2016 11:10	E112
9	550000010	Time when doctor examination or medication dispensing completed	9/3/2016 11:12	E112
10	550000010	Time of medication dispensing	9/3/2016 12:14	E112
11	550000010	Receipt recording/Time of receipt entry	9/3/2016 12:19	E112
12	550000011	Time of service registration/admission	10/2/2016 13:43	R42
13	550000011	Time of history taking	10/2/2016 14:03	R42
14	550000011	Time when history taking completed	10/2/2016 14:05	R42
15	550000011	Time of doctor examination	10/2/2016 14:43	R42
16	550000011	Time sent to laboratory	10/2/2016 14:43	R42
17	550000011	Time when doctor examination or medication dispensing completed	10/2/2016 14:44	R42
18	550000011	Time of medication dispensing	10/2/2016 14:49	R42
19	550000011	Time of laboratory result receipt	10/2/2016 14:58	R42
20	550000011	Receipt recording/Time of receipt entry	10/2/2016 14:59	R42
21	550000011	Time of laboratory report issuance	10/2/2016 15:00	R42
22	550000011	Time of service registration/admission	1/3/2016 20:00	K30
23	550000011	Time of emergency room admission	1/3/2016 20:04	K30
24	550000011	Time of history taking	1/3/2016 20:06	K30
25	550000011	Time when history taking completed	1/3/2016 20:07	K30
26	550000011	Receipt recording/Time of receipt entry	1/3/2016 20:08	K30
27	550000011	Time when doctor examination or medication dispensing completed	1/3/2016 20:08	K30
28	550000011	Time when emergency room examination completed	1/3/2016 20:14	K30
29	550000011	Time of medication dispensing	1/3/2016 20:30	K30
30	550000011	Time of service registration/admission	20/4/2016 08:01	E119
31	550000011	Time sent to laboratory	20/4/2016 08:05	E119
32	550000011	Time of laboratory result receipt	20/4/2016 08:05	E119
33	550000011	Time of history taking	20/4/2016 08:18	E119
34	550000011	Time when history taking completed	20/4/2016 08:19	E119
35	550000011	Time of laboratory report issuance	20/4/2016 09:04	E119
36	550000011	Time of doctor examination	20/4/2016 11:11	E119
37	550000011	Time when doctor examination or medication dispensing completed	20/4/2016 11:14	E119
38	550000011	Time of medication dispensing	20/4/2016 11:28	E119
39	550000011	Receipt recording/Time of receipt entry	20/4/2016 11:33	E119
40	550000012	Time of service registration/admission	10/2/2016 12:13	U6579
41	550000012	Time of history taking	10/2/2016 12:14	U6579

Fig. 10. Data after automatic formatting using the AI model.

The system generated an event log containing 128,917 events from 10,217 unique cases, identifying all the 14 activities. This transformation perfectly preserved temporal relationships and activity sequences.

C. Process Mining Tool Integration Testing

The generated event log was subjected to compatibility testing using Disco for validating its quality and practical usability. Data were imported into Disco without any errors or data loss (Fig. 11).

The screenshot shows the Disco software interface with the imported event log data. The table displays columns: CaseID, Activity, and Timestamp. The data is organized into a list of events, each with a unique CaseID and a corresponding Activity and Timestamp.

CaseID	Activity	Timestamp
1 550000010	Time of service registration/admission	2016-03-09 08:03:00
2 550000010	Time sent to laboratory	2016-03-09 08:05:00
3 550000010	Time of history taking	2016-03-09 08:05:00
4 550000010	Time of laboratory result receipt	2016-03-09 08:19:00
5 550000010	Time of laboratory report issuance	2016-03-09 09:21:00
6 550000010	Time when history taking completed	2016-03-09 10:54:00
7 550000010	Time of doctor examination	2016-03-09 11:10:00
8 550000010	Time when doctor examination or medication dispensing completed	2016-03-09 11:12:00
9 550000010	Time of medication dispensing	2016-03-09 12:14:00
10 550000010	Receipt recording/Time of receipt entry	2016-03-09 12:19:00
11 550000011	Time of service registration/admission	2016-02-10 13:43:00
12 550000011	Time of history taking	2016-02-10 14:03:00
13 550000011	Time when history taking completed	2016-02-10 14:05:00
14 550000011	Time of doctor examination	2016-02-10 14:41:00
15 550000011	Time sent to laboratory	2016-02-10 14:43:00

Fig. 11. Data import into Disco.

The Disco analysis revealed clear and logical process maps. The system could identify 3,068 process variants, reflecting the diversity of real patient care processes. Performance metrics analysis showed a median case duration of 3.5 h and mean case duration of 12 days, consistent with the characteristics of medical processes. The statistical overview within Disco is shown in Fig. 12.

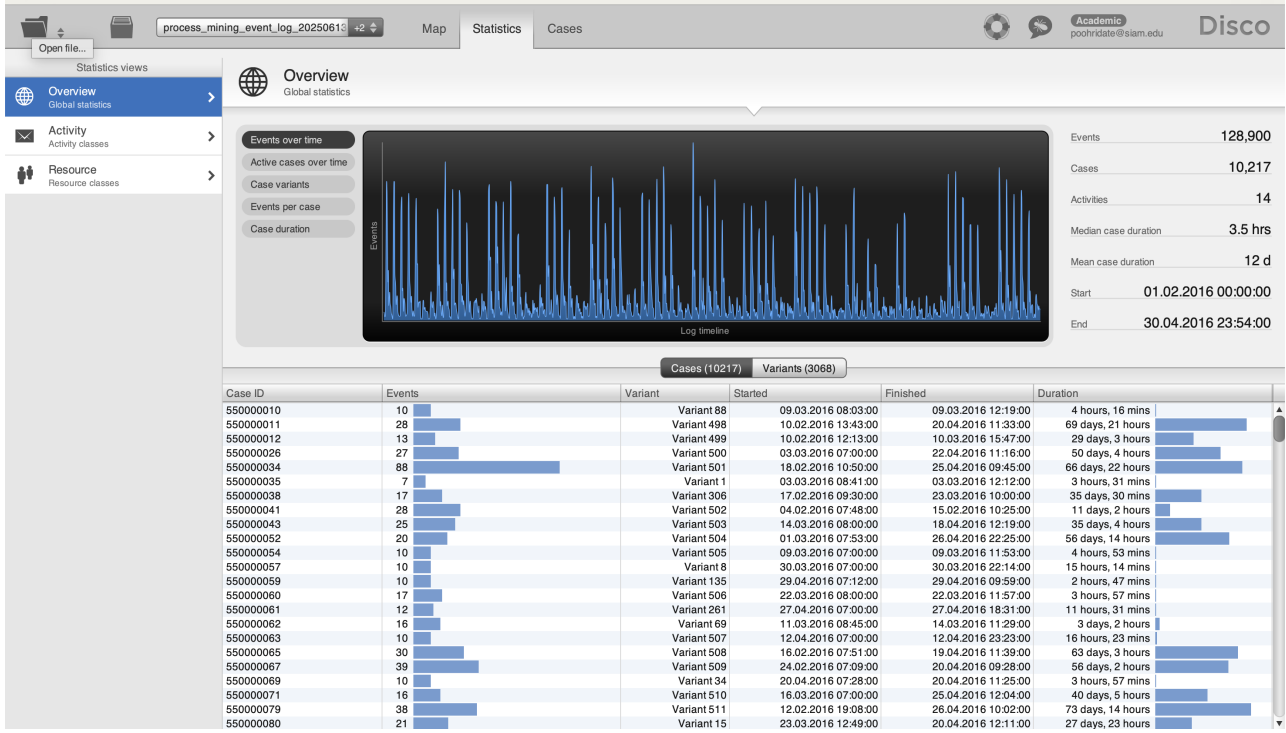


Fig. 12. Basic statistical data from Disco.

Process flow visualization in Disco clearly revealed bottlenecks and process flow patterns, particularly the identification of time-consuming activities and high-frequency paths. These results confirmed that the event log could be immediately used for process mining, validating the system as a viable and robust preparation tool.

D. Correctness and Integrity Validation

Beyond performance metrics, the correctness of the generated event log was validated in two parts to ensure its practical utility.

First, the process model discovered in Disco (Figs. 13 and 14) was presented to a hospital data analyst familiar with the outpatient workflow. The analyst confirmed that the primary pathways, activities, and identified bottlenecks logically and accurately represented the actual process.

This offered crucial qualitative validation of the data transformation correctness of the proposed system.

Second, a data integrity check was performed by comparing the key aggregate statistics between the raw data and the final event log. As shown in Table II, the counts for unique cases and total events matched perfectly. This indicated that no data was lost or erroneously duplicated during automated transformation, ensuring the structural integrity and completeness of the resulting event log.

TABLE II. CORRECTNESS AND INTEGRITY VALIDATION

Metric	Raw data count	Generated log count	Match
Total Unique Cases	10,217	10,217	Yes
Total Events	128,917	128,917	Yes

VI. CONCLUSION

A local LLM-based system was developed for automated event log data preparation for use in process mining, and its performance was evaluated. This system addressed existing problems in process mining such as significant time consumption, security risks from using online AI services, and technical barriers for novice analysts. The developed system used the Ollama framework with the Llama 3.1 model operating completely within the internal organizational environment.

A. Research Summary

The system successfully achieved all the research objectives. It was tested with real data obtained from a hospital in Thailand. The dataset contained 128,917 events, which were transformed into standard event logs by the developed system in merely 30.74 s (processing rate of 4,194 events per second). It successfully handled Thai data mixed with English technical terminologies while maintaining translation consistency via an efficient caching mechanism.

This indicated that the developed system has fundamental advantage over traditional cloud-based approaches, shifting the paradigm from the potential of LLMs to their practical, secure, and scalable applications. Its ability of process 128,917 sensitive hospital events with zero external data transmission makes it a viable solution for organizations that have strict data privacy regulations, thereby directly mitigating data leakage risks inherent in cloud-based approaches. By completing such a complex task in a single 30-s run, the system overcomes the critical scalability barriers of token limits and

unpredictable costs associated with public APIs—a major challenge in previous studies. Its usability was confirmed via its seamless import and coherent generation of a process model in Disco (Figs. 13 and 14). This further confirmed that the system ensured data integrity and delivered excellent results compared with those obtained from error-prone manual methods; it was also more secure than the existing AI-assisted cloud workflows. These findings confirm that the developed system is not only unique but also more robust, secure, and enterprise ready than the existing methods.

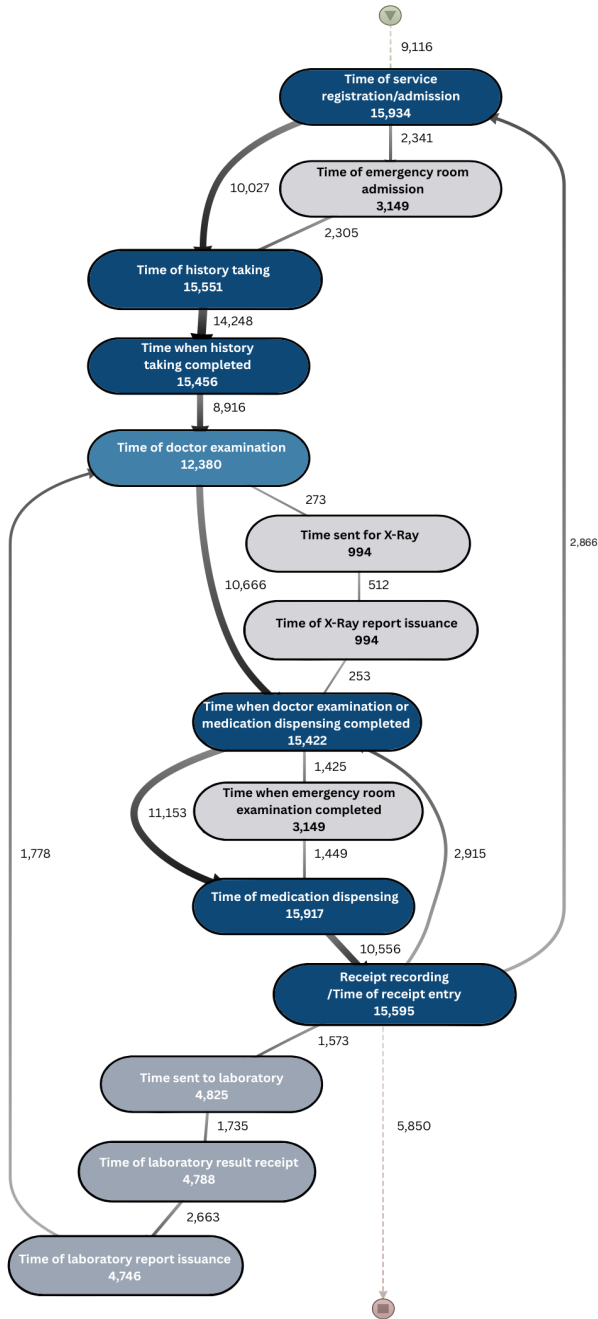


Fig. 13. Frequency pattern from the Fuzzy Miner algorithm.

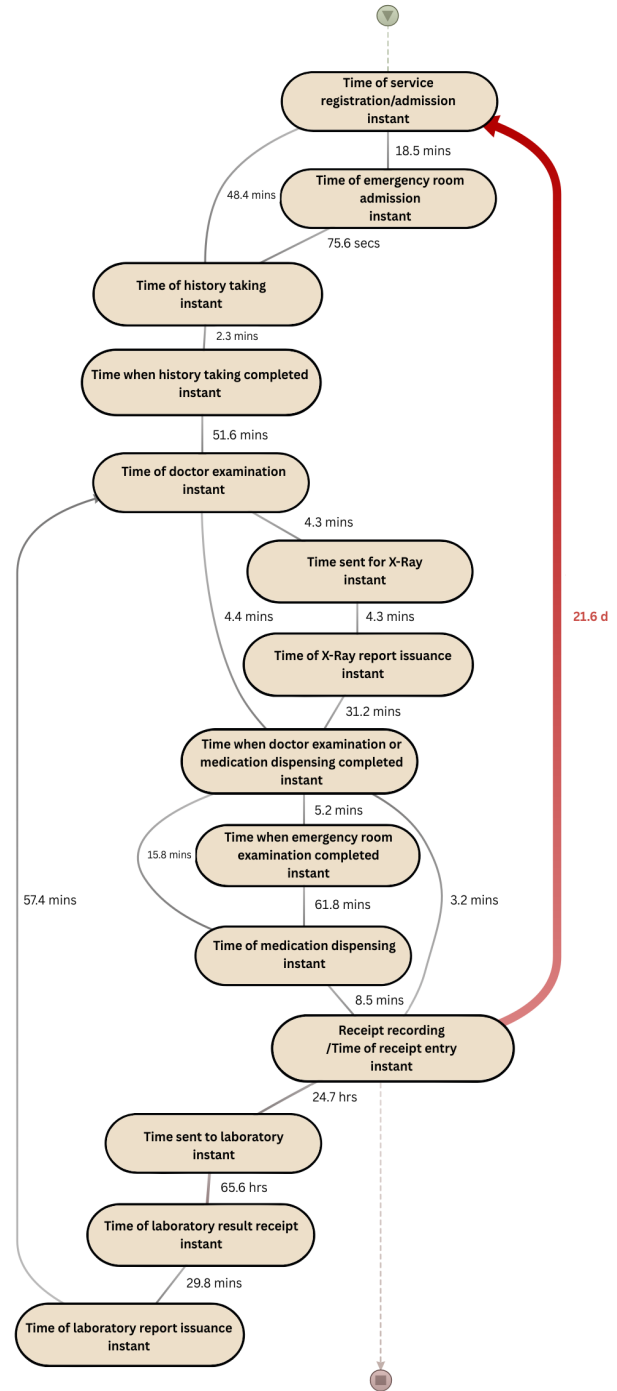


Fig. 14. Time pattern from the Fuzzy Miner algorithm.

This system also considerably impacts process mining. It democratizes access to process mining by making the technology more accessible for small and medium-sized organizations by reducing data preparation time from 60–90% of the project timeline to merely a few minutes. It also reduces errors associated with manual data preparation and extends process mining to global contexts by enabling multilingual data support.

B. Recommendations

In future, data pattern detection algorithms must be developed to yield highly accurate and flexible results using advanced machine learning techniques. The development should focus on enhancing the capabilities of the system to handle complex structured data, support other file formats, and expand testing scenarios to other domain data such as banking, manufacturing, and retail. Its performance should also be compared with other tools.

For practical application, organizations must first experiment with small-scale, nonsensitive data; develop teams that understand both process mining and AI technology; establish AI usage guidelines and standards; and invest in IT infrastructure that supports internal organizational AI processing. In addition, educational institutions should incorporate AI-assisted data preparation content into curricula, promote collaborative research between institutions and industry, and establish specialized research centers.

A limitation of this study is that a direct performance benchmark against other data preparation tools or cloud-based AI solutions has not been established. Future work can address this issue by facilitating such a comparison to quantitatively position the performance of local LLM-based systems against established alternatives on various hardware configurations.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Poohridate Arpasat: Conceptualization, Methodology, Software Development, Data Analysis, and Writing—Original Draft. Wichian Premchaiswadi: Supervision, Validation, and Writing—Review & Editing. All authors had approved the final version.

REFERENCES

- [1] H. M. Marin-Castro and E. Tello-Leal, "Event log preprocessing for process mining: A review," *Applied Sciences*, vol. 11, no. 22, 10556, 2021.
- [2] N. Martin, D. A. Fischer, G. D. Kerpedzhiev *et al.*, "Opportunities and challenges for process mining in organizations: Results of a Delphi study," *Business & Information Systems Engineering*, vol. 63, no. 5, pp. 511–527, 2021.
- [3] W. M. P. van der Aalst, *Process Mining: Data Science in Action*, 2nd ed. Berlin: Springer, 2016.
- [4] B. C. Das, M. H. Amini, and Y. Wu, "Security and privacy challenges of large language models: A survey," *ACM Computing Surveys*, vol. 57, no. 6, pp. 1–39, 2025.
- [5] C. W. Günther and W. M. P. van der Aalst, "Fuzzy mining—adaptive process simplification based on multi-perspective metrics," in *Proc. International Conference on Business Process Management*, 2007, pp. 328–343.
- [6] N. Martin, G. van Houdt, and G. Janssenswillen, "DaQAPO: Supporting flexible and fine-grained event log quality assessment," *Expert Systems with Applications*, vol. 191, 116274, 2022.
- [7] D. Fahland, "Extracting and pre-processing event logs," arXiv preprint, arXiv:2211.04338, 2022.
- [8] A. Berti and W. M. P. van der Aalst, "Leveraging Large Language Models (LLMs) for process mining," arXiv preprint, arXiv:2307.12701, 2023.
- [9] F. K. Parast, C. Sindhav, S. Nikam *et al.*, "Cloud computing security: A survey of service-based models," *Computers & Security*, vol. 114, 102580, 2022.
- [10] A. M. Ghosh and K. Grolinger, "Edge-cloud computing for internet of things data analytics: Embedding intelligence in the edge with deep learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2191–2200, 2021.
- [11] B. Estrada-Torres, A. del-Rio-Ortega, and M. Resinas, "Mapping the landscape: Exploring large language model applications in business process management," in *Proc. International Conference on Business Process Modeling, Development and Support*, 2024, pp. 22–31.
- [12] M. F. Sani, M. Sroka, and A. Burattin, "LLMs and process mining: Challenges in RPA: Task grouping, labelling and connector recommendation," in *Proc. International Conference on Process Mining*, 2023, pp. 379–391.

Copyright © 2025 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).