The International Journal of Digital Accounting Research
Vol. 24, 2024, pp. 119-140
ISSN: 2340-5058

Applying deep learning to detect abnormal event log traces: a non-rule-based framework

Yunsen Wang. Montclair State University, USA, wangyu@montclair.edu

Tiffany Chiu. The State University of New York at New Paltz, USA, chiut@newpaltz.edu

Miklos A. Vasarhelyi. Rutgers, the State University of New jersey, USA, miklosv@business.rutgers.edu

Abstract. Process mining is an efficient method that can analyze the full population of transactions using the event log of business processes. Conventional rule-based process mining techniques can detect anomalies; however, it tends to trigger a large number of false alarms. To improve the efficiency of anomaly detection using process mining, this study adopts a deep learning-based classification approach to detect anomalies in the traces of event logs. This approach contributes to the literature by proposing a non-rule-based process mining technique based on deep learning. Results demonstrate that the proposed non-rule-based process mining method can help auditors focus on transactional anomalies.

Keywords: Process mining, deep learning, anomaly detection, fraudulent activities.

JEL Code: M41

Acknowledgement

The authors are thankful for the financial support from the Rutgers, the State University of New Jersey.

DOI: 10.4192/1577-8517-v24_5

Submitted May 2024
Accepted November 2024

1. INTRODUCTION

Auditing plays a critical role in ensuring the integrity and transparency of financial reporting in organizations. Traditional analytical procedures conducted by auditors often rely on sampling techniques and manual inspection of documents, which can be time-consuming, error-prone, and may not capture the entire spectrum of deviations from compliance or inefficiencies within business processes. Process mining offers auditors a powerful toolkit for analyzing, monitoring, and enhancing organizational processes. To perform process mining analysis, four variables from the event log need to be extracted from the system: (1) Process Instance (Transaction) (2) Activity (Event), (3) Resource (the employee who creates the activity), and (4) Timestamp. For example, if Peter creates sales order 123 on April 12, 2024 at 11:00 AM, then "sales order 123" is the Process Instance, "create sales order" is the Activity, "Peter" is the Resource, and "April 12, 2024 at 11:00 AM" is the Timestamp. A variant in process mining is a group of process instances that have an identical pattern, which is called the trace. For example, process instances A and B can be group into the same variant if they both have the following trace: "Order created: Standard order -> Order adjusted: Goods Issue Date -> Order adjusted: Confirmed Quantity -> Order adjusted: Loading Date -> Goods issue -> Invoice created: Invoice -> Invoice posted: Accounts receivable -> Invoice cleared." The trace length is the number of steps that the process takes to complete. For the above example, the trace length is eight.

Jans et al. (2014) demonstrated the application of process mining in the auditing field and identified transactional anomalies undetected by the internal auditors with the conventional analytical procedures. Chiu et al. (2019) used process mining to assess the effectiveness of internal controls by classifying variants into standard and non-standard categories. However, the adoption of process mining in the audit practice presents several challenges. One of the concerns is the large number of process instances being classified as exceptions by process mining techniques. Although according to SAS No.99, fraud is an important consideration in audit analytical procedures (AICPA, 2002), requiring auditors to investigate as many exceptions as possible, yet most of the exceptions raised by the existing process mining techniques could turned out to be the "false alarms." As a result, auditors with limited time and resources allocated to an engagement could not investigate all the exceptions generated by existing process mining techniques.

Most existing process mining techniques are rule-based classification systems (Jans et al., 2014; Chiu and Jans, 2019). Rule-based process mining techniques output a large number of variants that do not comply with business rules, most of these exceptions turn out to be low-risk variants based on the organization's business rules. There is limited research focusing on developing non-rule-based process mining techniques. To address the practical question of adopting process mining in audit practices, this study proposes a non-rule-based process mining technique by employing deep learning for detecting event log anomalies. Specifically, the deep learning architecture that is used in this study is called the sequence to sequence (Seq2Seq) model, which can process the sequential data. In the accounting field, many applications of deep learning in anomaly detection focus on the data from general ledger and journals (Sun, 2019). This study proposes a framework of detecting abnormal event logs using a non-rule-based process mining technique based on deep learning. To evaluate the effectiveness of the proposed framework, this study uses a real-world event log data and demonstrates how anomalies can be detected by the non-rule-based process mining technique.

To reduce the high rate of false alarms in anomaly detection generated by conventional rule-based process mining techniques, this study introduces a framework incorporating deep learning techniques to identify the abnormal event log traces. Deviating from rule-based methodologies, this approach offers a non-rule-based alternative. The findings indicate that the proposed framework aids auditors in focusing on transaction anomalies, thus improving the effectiveness of anomaly detection processes. By incorporating deep learning, the study offers a promising approach for enhancing the accuracy and efficiency of anomaly detection in process mining, thereby expanding the utility of this methodology in audit practice.

The remainder of this paper is organized as follows. Section II reviews the existing literature of the application of process mining in the auditing field. Section III proposes a framework of the non-rule-based process mining for abnormal event log trace detection. Section IV presents the evaluation of the proposed framework and compares its performance with conventional rule-based process mining for anomaly detections. Section V concludes the study and discusses the contributions.

2. LITERATURE REVIEW

2.1. Process mining challenges

Process mining of event logs is a method for understanding an organization's business processes. It has been developed over the last decade by computer scientists and statisticians in collaboration with leading corporations, such as SAP and Phillips (Jans et al., 2014). Process mining provides a visualized solution to present the complex business processes and facilitates the comparison of the actual processes against the designed processes (van der Aalst, 2011). The existing process mining techniques involve three phases: discovery, conformance, and enhancement. It benefits the data analysts in 1) enhanced visibility with a comprehensive view of organizational processes, and discovering inefficiencies and compliance violations; 2) real-time monitoring and detecting anomalies, reducing the risk of fraud; and 3) tracing the origins of process deviations, aiding in identifying underlying issues. Since its emergence, process mining has been widely adopted in many organizations in various fields, such as industrial engineering, healthcare, and network security (Thiede et al., 2018).

Auditing, as a vital aspect of ensuring firms' financial reporting integrity, has traditionally relied on manual techniques to detect errors and inefficiencies. The emergence of process mining presents an opportunity to revolutionize auditing by leveraging data-driven analysis. Process mining has been introduced to the auditing field for testing the effectiveness of internal controls, with a demonstration of finding transactional anomalies that were not detected by the internal auditors using the conventional analytical procedures (Jans et al. 2014). Since then, accounting information systems (AIS) research has been conducted to apply process mining techniques to the auditing field (Chiu et al. 2019). Studies by Jans et al. (2014), Chiu et al. (2019), and Wang et al. (2020) demonstrated that process mining techniques can be applied to internal control assessment, the identification of standard and non-standard variants, and can be integrated to assist the compliance of the new revenue recognition standards. Chiu and Jans (2019) developed a framework of identifying standard and non-standard variants and process instances. Chiu et al. (2020) extended the application of process mining to fraud detection. For training professionals and students in the use of process mining, Hawkins et al. (2023) developed an event log generation tool for the order-to-cash process, creating synthetic datasets closely resembling real-world audit practices. These

datasets incorporate various elements such as segregation of duties issues, internal control violations, operational inefficiencies, and fraudulent behavior.

Anomaly detection through process mining is being used in various ways and for different purposes. Ghionna et al. (2008) introduced a cluster-based method to identify outliers using process mining data. Bezerra et al. (2009) proposed ProM tools, addressing the delicate balance between flexibility and security in Process Aware Information Systems (PAISs), thereby enhancing competitiveness while upholding security standards. Myers et al. (2018) utilized process mining to prevent cyber-attacks by detecting anomalies through conformance checks. Saraeian and Shirazi (2020) focused on analyzing manufacturing processes using process mining techniques. Furthermore, Tavares et al. (2018) introduced an online anomaly detection method for business processes, leveraging density-based clustering to detect outliers in real-time. Tavares and Junior (2021) explored meta-learning-based process mining, offering innovative avenues for anomaly detection. Sarno et al. (2020) proposed an integrated approach combining process mining with fuzzy rule-based classification and multi-attribute decision making, achieving remarkable accuracy in fraud detection within ERP systems. Vitale et al. (2023) employed autoencoders for anomaly detection in industrial Internet of Things settings. While these approaches demonstrate promising results, expanding and maintaining robust rule-based process mining techniques for anomaly detection can still be complex, especially as organization's business rules evolve over the period. Moreover, false alarms may arise if rules are not meticulously defined, potentially leading to inefficiency. As the process mining field continues to evolve, addressing these challenges will be essential to ensure process mining technique's reliability and efficiency in detecting anomalies.

2.2. Deep learning applications

The development of AI based on deep learning has made remarkable progress in recent years, including computer vision (e.g., autonomous driving) and natural language processing (e.g., ChatGPT). Large language models-based AI, such as ChatGPT, leverage a specific deep learning architecture, Seq2Seq models, that can take in and predict sequential data. The Seq2Seq deep learning models have emerged as powerful tools for solving a wide range of sequence-related tasks (Keneshloo et al., 2019), including machine translation, text summarization, speech recognition, etc. These models leverage recurrent neural (RNNs) networks

(Rodriguez, 1999), long short-term memory (LSTM) networks (Lindemann et al., 2021), and increasingly transformer architectures (Han et al., 2021) to learn complex mappings from input sequences to output sequences. The Seq2Seq models consist of an encoder and a decoder network. The encoder processes the input sequence and generates a fixed-size context vector, capturing the input's semantic information. The decoder then uses this context vector to generate the output sequence step by step.

There are wide applications of deep learning-based Seq2Seq models in natural language processing. The Seq2Seq models have been applied to machine translation. Seq2Seq models have improved machine translation tasks by learning to translate text from one language to another without the need of handcrafted rules or alignments (Behrmann et al., 2022). In text summarization, Seq2Seq models are widely used for generating concise summaries of long documents or articles, capturing the essential information while discarding redundant details (Shi et al., 2021). Seq2Seq models are also employed in speech recognition systems to convert spoken language into text, enabling applications such as virtual assistants and voice-controlled devices (Dong et al., 2018). To meet the demand of robot dialogues, the Seq2Seq models facilitates conversations by learning to generate appropriate responses based on input utterances, fostering natural and engaging interactions with users (Saluja et al., 2024).

The recent advancements of Seq2Seq models include pre-training models on large datasets before fine-tuning on specific tasks, such as ChatGPT, showing promise in improving performance and generalization (Wu et al., 2023). As Seq2Seq models become increasingly pervasive, addressing ethical concerns, such as data privacy, security, and societal impact, is paramount to fostering responsible AI development and deployment (Radanliev et al., 2024). While significant progress has been made in the deep learning architecture design, applications, and performance, the application of process mining in auditing research should also incorporate the power of Seq2Seq models, addressing audit practitioners' needs of a more efficient process mining tool. The Seq2Seq model is particularly well-suited for event logs due to its ability to handle sequential data, making it ideal for capturing the order and dependencies of activities over time. Its self-training architecture allows the model to learn complex patterns from the event logs without requiring extensive

manual labelling. This capability ensures that the model can adapt and improve over time, effectively identifying subtle anomalies and variations in business processes.

Building on the findings of Chiu and Jans (2019), this study takes a step further to integrate deep learning to enhance anomaly detection using process mining. The process mining technique proposed in this study is different from the approach taken by Bezerra et al. (2009), where abnormal variants are filtered using a fitness criterion based on adherence to predefined business rules. In addition, different from Sarno et al. (2020), who rely on experts to evaluate rules-based systems for credit card application anomaly detection, this study employs AI-based machine learning methods that continuously learn from event logs. In contrast to the prevailing practice of using rule-based process mining techniques for anomaly detection, the proposed AI-based process mining technique offers a more adaptive approach. While rule-based systems are straightforward and cost-effective, they struggle with adaptability and ambiguity. Non-rule-based process mining techniques, on the other hand, demonstrate dynamic adaptability and enable auditors to handle complex scenarios.

3. RESEARCH METHODOLOGY

Following the design science research paradigm (Hevner et al. 2004), this study first identifies a relevant business problem and designed an artifact. Then, we evaluate the proposed artifact by rigorous research methods, Finally, we present our contributions from both technology and business-oriented perspectives. Therefore, this study proposes a framework for incorporating deep learning-based Seq2Seq architectures into process mining for anomaly detection in audit practices. Furthermore, it evaluates the proposed framework by employing the Seq2Seq model to detect anomalies by comparing the outcomes with conventional process mining and machine learning anomaly detection methods.

3.1. Data creation process

In the context of analytical procedures utilizing process mining, auditors start with extracting event logs from ERP systems. For instance, if a client operates its daily business with an ERP system that records economic events during the business processes, an event log can be constructed by aggregating data from the tables in the accounting information system. These tables include Purchase Orders, Goods Receipts, Invoice Receipts, and Cash Payment tables from the procure-to-pay (P2P)

transaction cycles, as well as Sales Orders, Shipments, Charge Invoices, and Cash Payment tables from the order-to-cash (O2C) transaction cycles (illustrated in Figure 1). In the case of the O2C event log, essential details such as transaction ID/sales order number, timestamps, operators/employees (referred to as resources in process mining research), and transaction values, must be extracted and consolidated through SQL queries. The event log dataset, comprising the complete set of transactions, should incorporate at least the following variables: transaction ID, business event/activity, timestamp, employee/resource information, and transaction values.

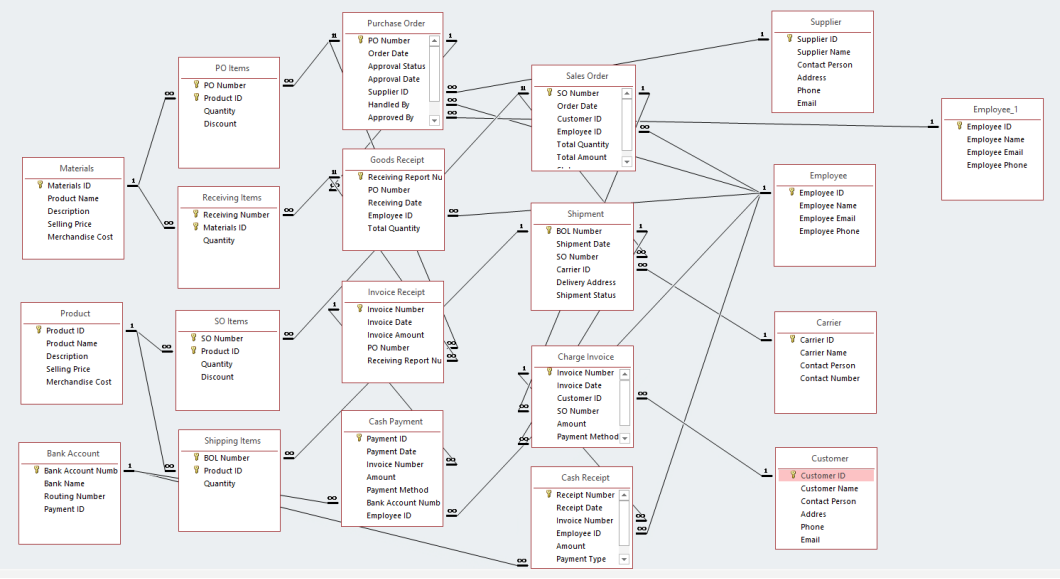

Figure 1. Sample database for ERP system

We first use a synthetic dataset from the process mining case study provided by the EY Academic Resource Center (EYARC) to develop the framework and apply it to a real-world event log dataset. This synthetic dataset comprises 1,000 transactions, encompassing 9,734 activities. These transactions occurred between January 1, 2021, and March 30, 2021. The total sales revenue for the fiscal quarter amounted to $1,073,855.05. For each transaction, activities are ordered based on timestamps. Subsequently, the dataset is organized into process instances. As illustrated in Table 1 Panel A, process instances are grouped into variants if they present the same trace. Panel B shows the activity encoding, presenting the activity each number represents. For instance, Variant 1 shows 529 process instances have the following trace: Create Digital Purchase Order -> Credit Approved -> Create Picking Ticket

-> Record Picking of Inventory -> Create Shipping Documents -> Review and Approve Sales Invoice -> Email Sales Invoices to Customer -> Receive Payment -> Print Bank Deposit Slip -> Match Deposit Slip and Bank Receipt. More than half of the dataset follow this particular trace, suggesting that this variant should be considered a standard business process. All other variants are non-standard variants but may also be pre-defined by business rules. For example, Variant 2 shows the trace as: Create Digital Purchase Order -> Credit Denied. Although this variant is not the most frequent variant, it should not be labeled an exception because the pre-defined business rules state that if a customer's credit check fails, the company must reject the customer's purchase order. Conventional process mining techniques must rely on the organization's internal policies, such as standard business rules, and then determine which variant or transaction are anomalies that require auditors' further investigations.

Panel A

Variant and Variant Trace

Variant Number	Variant Trace	Frequency
1	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 9 -> 10 -> 11 -> 12	529
2	1 -> 3	79
3	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 9	75
4	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 9 -> 10 -> 14 -> 11 -> 13 -> 12	74
5	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 9 -> 10 -> 11 -> 13 -> 14 -> 12	64
6	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 9 -> 10 -> 14 -> 13 -> 11 -> 12	50
7	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 9 -> 10 -> 13 -> 11 -> 14 -> 12	39
8	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 9 -> 10 -> 11 -> 13 -> 12 -> 14	32
9	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 10 -> 13 -> 11 -> 14 -> 12	23
10	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 9 -> 10 -> 13 -> 11 -> 12 -> 14	19
11	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 10 -> 11 -> 13 -> 14 -> 12	12
12	1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 9 -> 10 -> 11	2
13	1 -> 2 -> 4 -> 5 -> 6 -> 9 -> 7 -> 10 -> 11 -> 12	1
14	1 -> 2 -> 4 -> 5 -> 6 -> 9 -> 7 -> 10 -> 14 -> 13 -> 11 -> 12	1

Panel B

Activity Encoding

1	Create Digital Purchase Order
2	Credit Approved
3	Credit Denied
4	Create Picking Ticket
5	Record Picking of Inventory
6	Create Shipping Documents
7	Review and Approve Sales Invoice
8	Review and Reject Sales Invoice
9	Email Sales Invoices to Customer
10	Receive Payment
11	Print Bank Deposit Slip
12	Match Deposit Slip and Bank Receipt
13	Shred Remittance
14	Scan and Save Remittance

Table 1. Variant trace and activity encoding

3.2. Framework

This study proposes a non-rule-based process mining method, which can detect event log anomalies efficiently. The proposed method enables auditors to apply process mining techniques in their audit work even when the information of standard business rules is incomplete or missing, or when conventional process mining yields complex results with a large number of exceptions. The framework outlined below demonstrates the stages involved in identifying abnormal variants. An overview of the framework is depicted in Figure 2. In general, there are three stages of the application: 1) Using variant trace length percentiles to detect abnormal variants, 2) Using Seq2Seq autoencoder to train and calculate reconstruction errors; and 3) Using Z-scores to detect variants with abnormal reconstruction errors.

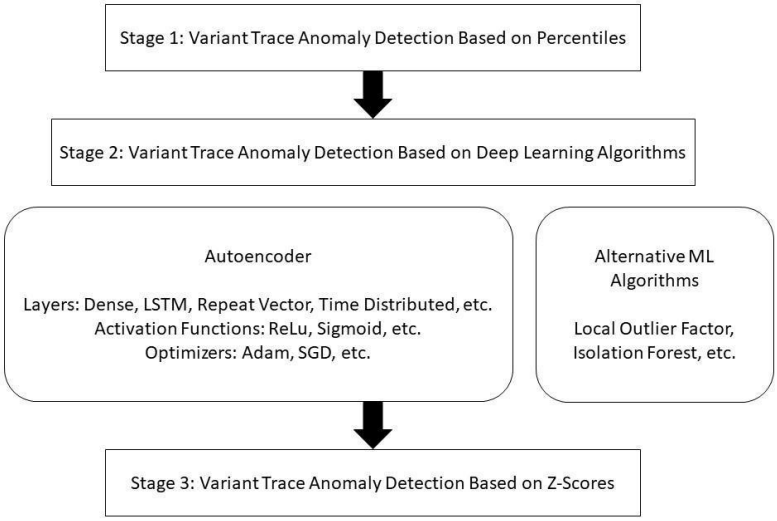

Figure 2. Framework

Stage 1: Using variant trace length percentiles to detect abnormal variants

Some variant traces extend significantly in length, exceeding 100 activities to complete a transaction. Such transactions are considered anomalies as it is expected that a normal business process should be completed within a limited number of steps. A complex and extensively long business process could indicate suspicious transactions. Therefore, the first step in this framework involves detecting variant anomalies based on the percentile of the number of steps included in each variant. For instance, variants with trace lengths that are greater than the 95th percentile of the entire population are identified as anomalies. When applying the framework to audit practice, auditors can select the percentiles ranging from 90th to 99th, depending on the availability of their resources and allocated time for the audit engagement. Choosing the 90th as the threshold would yield more anomalies than 99th.

Stage 2: Use seq2seq autoencoder to train and calculate reconstruction errors

Next, for the remaining variants, this study uses Python Keras, a deep learning platform, to develop a Seq2Seq autoencoder to conduct unsupervised learning. The Seq2Seq autoencoder comprises two components: the encoder and the decoder. The encoder reduces the dimensions of the dataset from high to a fixed-length bottleneck, while the decoder reconstructs the dataset from the fixed-length

bottleneck. Since the unsupervised learning does not need labels for training the model, both the inputs and outputs of the Seq2Seq autoencoder are the same (i.e., variant traces). The Seq2Seq autoencoder learns these inputs through multiple layers of neural networks, including LSTM, Repeat Vector, Time Distribution Layers, and ultimately predicting the variant traces themselves. This process, termed reconstruction, enables us to calculate reconstruction errors. The reconstruction errors are calculated as the mean squared errors (MSEs) between the original and the predicted variant trace data, serving as the dissimilarity scores among the variant traces. In practice, auditors have the flexibility to customize the Seq2Seq autoencoder by selecting the number and type of layers, activation functions, and optimizer algorithms. The parameter tuning details are demonstrated in Table 2. This customization process allows auditors to develop the most suitable model for each event log data.

Parameter	**Option**
Layer Type	Dense, LSTM, Repeat Vector, Time Distribution, etc.
Activation Function	Linear, Relu, Sigmoid, etc.
Optimizer Algorithm	Stochastic Gradient Descent, Adam, etc.
Loss Function	Maximum Likelihood, Cross Entropy, Binary Cross Entropy, etc.
Learning Rate	0.0 - 1.0

Table 2. Seq2seq autoencoder fine tuning parameters

Stage 3: Use z-scores to detect variants with abnormal reconstruction errors

The third stage involves detecting abnormal variant traces using the Z-scores of the reconstruction errors. A common approach to identifying anomalies based on the Z-scores is to flag observations that exceed a threshold of 2. When applying this stage in the audit practice, auditors have the flexibility to adjust this threshold based on available resources and allocated time, opting for values between 2 (yielding more anomalies) and 3 (yielding fewer anomalies).

$$z = (X-\mu)/\sigma$$

Where, z is the Z-score, X is reconstruction error, μ and σ are the mean and standard deviation of the X.

4. FIELD STUDY AND EVALUATION

4.1. Descriptive statistics

The proposed framework is applied to a real-world event log dataset extracted from a large accounting firm's ERP system. The dataset comprises 17,196 process instances, encompassing 252,547 activities. These transactions occurred between April 28, 2014, and February 10, 2016. The dataset includes 66 types of activities, such as "Order Created: Standard," "Order Adjusted: Confirmed Quantity," "Goods Issue," "Invoice Posted: Accounts Receivable," and "Invoice Cleared." On average, a transaction takes approximately 46 days to complete. The statistics of the event log data and a sample of transaction trace are shown in Tables 3 and 4, respectively.

Process Instances	17,196
Activities	252,547
Activity Types	66
Variants	1,376
Start Date of the Event Log	4/28/2014
End Date of the Event Log	2/10/2016
Resource	23
Mean Process Instance Duration	46

Table 3. Descriptive statistics

Case ID	**Process Instance Traces**
0000031480-000010	Order created: Standard order -> Order adjusted: Confirmed Quantity -> Order adjusted: Overall status of credit checks -> Order adjusted: Release date of the document determined by credit management -> Order adjusted: Condition pricing unit -> Order adjusted: Confirmed Quantity -> Order adjusted: Item credit price -> Order adjusted: Item credit price -> Order adjusted: Overall status of credit checks -> Order adjusted: Total incompletion status of all items: Delivery -> Order adjusted: Net price -> Order adjusted: Confirmed Quantity -> Order adjusted: Overall status of credit checks -> Order adjusted: Status of static credit limit check -> Order adjusted: Goods Issue Date -> Order adjusted: Schedule line date -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Transportation Planning Date -> Order adjusted: Goods Issue Date -> Order adjusted: Schedule line date -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Transportation Planning Date -> Order adjusted: Confirmed Quantity -> Order adjusted: Order quantity in sales units -> Order adjusted: Order quantity in sales units -> Order adjusted: Overall processing status of document -> Order adjusted: Goods Issue Date -> Order adjusted: Schedule line date -> Order adjusted: Confirmed Quantity -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Next date -> Order adjusted: Requested delivery date -> Order adjusted: Transportation Planning Date -> Order adjusted:

	Shipping Point/Receiving Point -> Goods issue -> Invoice created: Invoice -> Invoice posted: Accounts receivable -> Invoice cleared
0000035150-000010	Order created: Standard order -> Order adjusted: Goods Issue Date -> Order adjusted: Schedule line date -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Next date -> Order adjusted: Requested delivery date -> Order adjusted: Transportation Planning Date -> Order adjusted: Goods Issue Date -> Order adjusted: Schedule line date -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Next date -> Order adjusted: Requested delivery date -> Order adjusted: Transportation Planning Date -> Order adjusted: Condition pricing unit -> Order adjusted: Item credit price -> Order adjusted: Item credit price -> Order adjusted: Total incompletion status of all items: Delivery -> Order adjusted: Net price -> Order adjusted: Goods Issue Date -> Order adjusted: Confirmed Quantity -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Transportation Planning Date -> Order adjusted: Schedule line date -> Order adjusted: Requested delivery date -> Order adjusted: Next date -> Order adjusted: Requested delivery date -> Goods issue -> Invoice created: Invoice -> Invoice posted: Accounts receivable -> Invoice cleared
0000035210-000010	Order created: Standard order -> Order adjusted: Goods Issue Date -> Order adjusted: Schedule line date -> Order adjusted: Condition pricing unit -> Order adjusted: Item credit price -> Order adjusted: Item credit price -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Next date -> Order adjusted: Requested delivery date -> Order adjusted: Total incompletion status of all items: Delivery -> Order adjusted: Transportation Planning Date -> Order adjusted: Net price -> Order adjusted: Condition pricing unit -> Order adjusted: Item credit price -> Order adjusted: Item credit price -> Order adjusted: Total incompletion status of all items: Delivery -> Order adjusted: Net price -> Goods issue -> Invoice created: Invoice -> Invoice posted: Accounts receivable -> Invoice created: Invoice -> Invoice posted: Accounts receivable -> Invoice cleared -> Invoice cleared

Table 4. Process instance traces sample

The process instances are grouped into variants based on their traces. Table 5 shows several examples of variants, and the numbers of process instance and activity included, and the average process duration (APD) in days. The most frequently occurred trace is Variant 1 (with 3,695 process instances included).

Variant Traces	**Number of Process Instances Included**	**Number of Activities Included**	**APD**
Order created: Standard order -> Order adjusted: Goods Issue Date -> Order adjusted: Confirmed Quantity -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Transportation Planning Date -> Goods issue -> Invoice created: Invoice -> Invoice posted: Accounts receivable -> Invoice cleared	3,695	10	56
Order created: Standard order -> Order adjusted: Goods Issue Date -> Order adjusted: Confirmed Quantity -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Transportation Planning Date -> Order adjusted: Goods Issue Date -> Order adjusted: Goods Issue Date -> Order adjusted: Schedule line date -> Order adjusted: Loading Date -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Transportation Planning Date -> Order adjusted: Transportation Planning Date -> Goods issue -> Invoice created: Invoice -> Invoice posted: Accounts receivable -> Invoice cleared	3,163	19	52
Order created: Standard order -> Goods issue -> Invoice created: Invoice -> Invoice posted: Accounts receivable -> Invoice cleared	2,331	5	48
Order created: Standard order -> Order adjusted: Goods Issue Date -> Order adjusted: Goods Issue Date -> Order adjusted: Schedule line date -> Order adjusted: Loading Date -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Transportation Planning Date -> Order adjusted: Transportation Planning Date -> Goods issue -> Invoice created: Invoice -> Invoice posted: Accounts receivable -> Invoice cleared	727	14	41
Invoice created: Inter-company -> Invoice posted: Revenue intercompany UK	618	2	0
Order created: Standard order -> Order adjusted: Goods Issue Date -> Order adjusted: Confirmed	594	18	5

Quantity -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Transportation Planning Date -> Order adjusted: Goods Issue Date -> Order adjusted: Goods Issue Date -> Order adjusted: Schedule line date -> Order adjusted: Loading Date -> Order adjusted: Loading Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Material Staging/Availability Date -> Order adjusted: Transportation Planning Date -> Order adjusted: Transportation Planning Date -> Goods issue -> Invoice created: Invoice -> Invoice posted: Accounts receivable			

Table 5. Example of variants

4.2. Framework application

In Stage 1, we utilize the 95th percentile as the criterion to identify abnormal variants; if the length of a variant exceeds that of 95 percent of its peers, it is flagged as abnormal. This step results in the identification of 69 abnormal variants, which are detailed in Table 6. Notably, the maximum process instances included in these variants are 2, indicating their rare occurrence. Additionally, the average process duration of 80.93 days suggests that these anomalous variants take a significantly longer time to complete, warranting further scrutiny. Moreover, the number of activities included in these variants ranges from 103 to 382, indicating their complexity and further reinforcing their suspicious nature.

	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
Activities	69	165.06	63.93	103	116	145	191	382
Process Instances	69	1.04	0.21	1	1	1	1	2
APD	69	80.93	22.53	16.44	70.61	80.42	97.55	125.42

Table 6. Outliers detected in stage 1

Moving to Stage 2, we build an autoencoder to identify abnormal activities. By leveraging a Seq2Seq autoencoder, we consider variant traces as the input dimension. We train the Seq2Seq autoencoder using the remaining 1,307 variants, calculating Mean Squared Errors (MSEs) to assess the reconstruction errors for each variant. Figure 3 illustrates the parameter specifications of the Seq2Seq autoencoder. In Stage 3, we apply Z-scores to the reconstruction errors, identifying abnormal variants with Z-scores surpassing 2. This stage reveals an additional 70 abnormal variants.

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, None, 67)	0	-
lstm (LSTM)	[(None, 64), (None, 64), (None, 64)]	33,792	input_layer[0][0]
repeat_vector (RepeatVector)	(None, 102, 64)	0	lstm[0][0]
lstm_1 (LSTM)	[(None, 102, 64), (None, 64), (None, 64)]	33,024	repeat_vector[0]… lstm[0][1], lstm[0][2]
input_layer_1 (InputLayer)	(None, 102, 67)	0	-
time_distributed (TimeDistributed)	(None, 102, 67)	4,355	lstm_1[0][0]

Figure 3. Sequence-to-sequence autoencoder

4.3. Effectiveness evaluation

Table 7 presents descriptive statistics for the 70 abnormal variants identified by the Seq2Seq autoencoder (from Stages 2 and 3). The majority of these variants only have one process instance included, suggesting they are rare and infrequent events. The identified variants have a prolonged duration, with an average completion time of 68.19 days, and have a large number of activities included, with an average of 79.71 activities.

	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
Activities	70	79.71	12.32	58	68	83	88	102
Process Instances	70	1.27	1.20	1	1	1	1	9
APD	70	68.19	35.15	9.37	47.35	68.35	87.43	168.52

Table 7. Outliers detected by autoencoder (in stages 2 and 3)

4.4. Benchmarking

We use two anomaly detection algorithms from machine learning, namely Local Outlier Factor (LOF) and Isolation Forest (IF), as benchmarks, comparing with the abnormal variants detected by the Seq2Seq autoencoder in Stage 2. LOF and IF have been widely used in the field of data mining and outlier detection (Cheng et al., 2019). They can also be easily implemented in the Python Scikit-learn library for benchmarking on the same platform as deep learning models (e.g., Keras and TensorFlow). Similar to the Seq2Seq autoencoder, LOF and IF can be employed in Stage 2 of the framework for outlier detection.

Local Outlier Factor identifies 141 outliers, while Isolation Forest detects 76. It is worth noting that the outliers detected by both algorithms share similar characteristics: they are rare occurrences, exhibit prolonged durations, and involve complex activities. In comparison to the autoencoder, these algorithms identify more outliers, potentially requiring auditors to allocate additional time and resources for further analysis. In audit practice, auditors can leverage these two algorithms to cross-validate the results obtained from the Seq2Seq autoencoder and gain insights into outliers detected by different methods.

	count	mean	std	min	25%	50%	75%	max
Activities	141	29.34	17.31	3	16	26	38	88
Process Instances	141	1.31	1.03	1	1	1	1	9
Avg Process Duration	141	59.56	48.59	0.01	16.76	57.12	95.36	315.35

Table 8. Outliers detected by local outlier factor

	count	mean	std	min	25%	50%	75%	max
Activities	76	74.00	18.39	22	59	82	88	102
Process Instances	76	1.14	0.93	1	1	1	1	9
Avg Process Duration	76	76.34	35.10	12.98	51.26	73.50	102.66	168.52

Table 9. Outliers detected by isolation forest

5. CONCLUSION AND DISCUSSIONS

This study proposes a framework of non-rule-based process mining for anomaly detection based on deep learning. The evaluation of the framework shows its performance compared with conventional rule-based process mining techniques

and the existing anomaly detection algorithms. This study contributes to the literature in the development of a non-rule-based process mining framework that is capable of analyzing the entire population of the variants and identifying abnormal variants. The proposed framework can be adopted by the auditors even if the client's business rules are missing or incomplete. Moreover, this study incorporates guidance that allows auditors to implement each stage of the framework in the audit practice.

This framework advances the work of Chiu et al. (2019) and Chiu et al. (2020) by offering a more refined approach to event log anomaly detection. While Chiu et al. (2019) primarily focused on classifying process variants into normal versus notable categories, this proposed framework takes a step further by specifically detecting a more concentrated group of abnormal variants. The framework aims to enhance the efficiency of process mining in auditing by reducing the likelihood of overwhelming auditors and diminishing the effectiveness of the process mining approach. This framework mitigates that inefficiency by filtering the data more effectively, allowing auditors to focus on the most relevant anomalies. By narrowing the scope to variants that are more likely to indicate issues, the framework streamlines the auditing process, making it more manageable and impactful for auditors.

However, a limitation arises from the lack of prior labels for the anomalies detected by the framework. We cannot verify with the data provider which variants are actual anomalies. As a result, measures of detection accuracy and sensitivity cannot be calculated. In future research, such limitations could be addressed by demonstrating how to simulate fraud in real-world event log data across various schemes, such as revenue recognition, foreign-related party transactions, account receivables, and inventory issues. Future study can also explore how the proposed method could identify fraud by considering variant traces, process durations, resources, employee information, transaction values, timestamps, and durations between events.

Despite the challenges, ongoing advancements in technology and methodology frameworks show that the application of process mining in audit practice is promising. By incorporating the proposed process mining technique alongside traditional auditing analytical procedures, auditors can enhance the effectiveness and efficiency of their audit work. This integration allows auditors to thoroughly

test the full population of their client's datasets, and at the same time minimizing false alarms through the adoption of the non-rule-based process mining technique.

6. REFERENCES

American Institute of Certified Public Accountants (AICPA). (2002). Consideration of Fraud in a Financial Statement Audit. Statement on Auditing Standards No. 99. New York, NY: AICPA. https://us.aicpa.org/content/dam/aicpa/research/standards/auditattest/downloadabledocuments/au-00316.pdf 2 December 2024

Behrmann, N., Golestaneh, S. A., Kolter, Z., Gall, J., & Noroozi, M. (2022). Unified fully and timestamp-supervised temporal action segmentation via sequence-to-sequence translation. In Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., & Hassner, T. (Eds.), *Computer Vision – ECCV 2022. Lecture Notes in Computer Science* (Vol. 13695, pp. 52–68). Springer, Cham. https://doi.org/10.1007/978-3-031-19833-5_4

Bezerra, F., Wainer, J., & van der Aalst, W. M. (2009). Anomaly detection using process mining. In *International Workshop on Business Process Modeling, Development and Support*, *29, 149-161.* https://doi.org/10.1007/978-3-642-01862-6_13

Cheng, Z., Zou, C., & Dong, J. (2019). Outlier detection using isolation forest and local outlier factor. In *Proceedings of the conference on research in adaptive and convergent systems*, 161-168. https://doi.org/10.1145/3338840.3355641

Chiu, T., & Jans, M. (2019). Process mining of event logs: A case study evaluating internal control effectiveness. *Accounting Horizons*, *33(3), 141-156.* https://doi.org/10.2308/acch-52458

Chiu, T., Wang, Y., & Vasarhelyi, M. A. (2020). The automation of financial statement fraud detection: a framework using process mining. *Journal of Forensic and Investigative Accounting, 12 (1), 86-108*. http://web.nacva.com/JFIA/Issues/JFIA-2020-No1-6.pdf 2 December 2024

Dong, L., Xu, S., & Xu, B. (2018). Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 5884-5888. IEEE. https://doi.org/10.1109/icassp.2018.8462506

Ghionna, L., Greco, G., Guzzo, A., & Pontieri, L. (2008). Outlier detection techniques for process mining applications. In *Foundations of Intelligent Systems: 17th International Symposium, Proceedings 17*, 150-159. https://doi.org/10.1007/978-3-540-68123-6_17

Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., & Wang, Y. (2021). Transformer in transformer. *Advances in neural information processing systems*, *34*, 15908-15919. https://doi.org/10.48550/arXiv.2103.00112

Hawkins, S. R., Pickerd, J., Summers, S. L., & Wood, D. A. (2023). The development of the process mining event log generator (PMELG) tool. *Accounting Horizons*, *37*(4), 85-95. https://doi.org/10.2308/horizons-2022-153

Hevner, Alan R., Salvatore T. March, Jinsoo Park, and Sudha Ram (2004). Design science in information systems research. *MIS quarterly*, *28*(1), 75-105. https://doi.org/10.1007/978-1-4419-5653-8_2

Keneshloo, Y., Shi, T., Ramakrishnan, N., & Reddy, C. K. (2019). Deep reinforcement learning for sequence-to-sequence models. *IEEE transactions on neural networks and learning systems*, *31*(7), 2469-2489. https://doi.org/10.1109/tnnls.2019.2929141

Jans, M., Alles, M. G., & Vasarhelyi, M. A. (2014). A field study on the use of process mining of event logs as an analytical procedure in auditing. *The Accounting Review*, *89*(5), 1751-1773. https://doi.org/10.2308/accr-50807

Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry*, *131*, 103498. https://doi.org/10.1016/j.compind.2021.103498

Myers, D., Suriadi, S., Radke, K., & Foo, E. (2018). Anomaly detection for industrial control systems using process mining. *Computers & Security*, *78*, 103-125. https://doi.org/10.1016/j.cose.2018.06.002

Radanliev, P., Santos, O., Brandon-Jones, A., & Joinson, A. (2024). Ethics and responsible AI deployment. *Frontiers in Artificial Intelligence*, *7*, 1377011. https://doi.org/10.3389/frai.2024.1377011

Rodriguez, P., Wiles, J., & Elman, J. L. (1999). A recurrent neural network that learns to count. *Connection Science*, *11*(1), 5-40. https://doi.org/10.1080/095400999116340

Saluja, K., Agarwal, S., Kumar, S., & Choudhury, T. (2024). Evaluating Performance of Conversational Bot Using Seq2Seq Model and Attention Mechanism. *EAI Endorsed Transactions on Scalable Information Systems*. https://doi.org/10.4108/eetsis.5457

Saraeian, S., & Shirazi, B. (2020). Process mining-based anomaly detection of additive manufacturing process activities using a game theory modeling approach. *Computers & Industrial Engineering*, *146*, 106584. https://doi.org/10.1016/j.cie.2020.106584

Sarno, R., Sinaga, F., & Sungkono, K. R. (2020). Anomaly detection in business processes using process mining and fuzzy association rule learning. *Journal of Big Data*, 7(1), 5. https://doi.org/10.1186/s40537-019-0277-1

Shi, T., Keneshloo, Y., Ramakrishnan, N., & Reddy, C. K. (2021). Neural abstractive text summarization with sequence-to-sequence models. *ACM Transactions on Data Science*, *2*(1), 1-37. https://doi.org/10.1145/3419106

Sun, T. (2019). Applying deep learning to audit procedures: An illustrative framework. *Accounting Horizons*, *33*(3), 89-109. https://doi.org/10.2308/acch-52455

Tavares, G. M., da Costa, V. G. T., Martins, V. E., Ceravolo, P., & Barbon Jr, S. (2018). Anomaly detection in business process based on data stream mining. In *Proceedings of the XIV Brazilian symposium on information systems*, 1-8. https://doi.org/10.1145/3229345.3229362

Tavares, G. M., & Junior, S. B. (2021). Process mining encoding via meta-learning for an enhanced anomaly detection. In *European Conference on Advances in Databases and Information Systems*, *1450,* 157-168. https://doi.org/10.1007/978-3-030-85082-1_15

Thiede, M., Fuerstenau, D., & Bezerra Barquet, A. P. (2018). How is process mining technology used by organizations? A systematic literature review of empirical studies. *Business Process Management Journal*, *24*(4), 900-922. https://doi.org/10.1108/bpmj-06-2017-0148

van der Aalst, W. M. P. (2011). Process Mining. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-19345-3

Vitale, F., De Vita, F., Mazzocca, N., & Bruneo, D. (2023). A process mining-based unsupervised anomaly detection technique for the industrial internet of things. *Internet of Things*, *24*, 100993. https://doi.org/10.1016/j.iot.2023.100993

Wang, Y., Chiu, T., & Chiu, V. (2020). Redesigning business process to comply with the new revenue recognition standard using process mining. *Journal of Emerging Technologies in Accounting*, *17*(1), 149-163. https://doi.org/10.2308/jeta-19-03-30-12

Wu, T., He, S., Liu, J., Sun, S., Liu, K., Han, Q. L., & Tang, Y. (2023). A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, *10*(5), 1122-1136. https://doi.org/10.1109/jas.2023.123618