QUEENSLAND UNIVERSITY OF TECHNOLOGY

IN10 Bachelor of Information Technology (Honours)

Zhipeng He
n10599070

Principal Supervisor: Dr. Chun Ouyang
Associate Supervisor: Dr. Catarina Moreira

School of Information Systems
Faculty of Science

October 31, 2021

Proposed Title

Project Title: Investigating the Impact of Event Logs on Deep Learning-based Process Prediction Performance

Project Type: Honours Research Project

Supervisory Team

Principal Supervisor: Dr. Chun Ouyang

Dr. Ouyang is a senior lecturer in the School of Information Systems and has supervised four PhD students to completion. She is an active and well-established researcher and the world top 21^{st} most cited scholar in Process Mining (according to Google Scholar). Built upon her expertise in process-oriented data mining, she has developed strong research interest in explainable predictive process analytics and submitted an ARC Discovery Project on the topic as the lead investigator. She is currently supervising four PhD students and one honours student as the principal supervisor and two PhD students as an associate supervisor.

Associate Supervisor: Dr. Catarina Pinto Moreira

Dr. Moreira is a lecturer in the School of Information Systems and holds the role of Deputy HDR Academic Lead. She is a Computer Scientist and passionate in investigating machine learning / deep learning models, and innovative human interactive probabilistic models for explainable AI. She has submitted a grant proposal on Persuasive and Causal Probabilistic Models for Explainable AI for ARC Discovery Early Career Researcher Award (DECRA). She is currently supervising three PhD students as the principal supervisor and three PhD students and one honours student as an associate supervisor.

Abstract

Business process predictive analytics exploit historical process execution logs, known as event logs, to generate predictions of running cases of a business process, such as next event or remaining time. In the state-of-the-art approaches, deep learning algorithms have attracted increasing attention and as a result deep learning-based prediction models become the mainstream of the research. Often encoding methods for event logs and neural network architectures have been considered as two factors that would impact models' prediction performance. In fact, an event log, as the input data for prediction, also plays an important role in the predictive pipeline and should not be overlooked. However, there is no recent research concerning with the potential influence of event logs on prediction performance. This thesis aims to investigate how different event logs affect the performance of deep learning-based process prediction models. We propose and implement a benchmark on two different encoding methods and three Long Short-Term Memory (LSTM) models with seven real-life event logs for predicting next activity, next resource and next interval time. Based on the above benchmark, this thesis explores and analyses some key characteristics of event logs and extracts findings on relationships between the characteristics of event logs and performance of process prediction models.

Keywords: Predictive Process Analytics; Deep Learning; Event Log.

Contents

Statement of original authorship

By submitting this thesis, I am aware of the University rule that a student must not act in a manner which constitutes academic dishonesty as stated and explained in the QUT Manual of Policies and Procedures. I confirm that this work represents my individual effort. I declare that it does not contain plagiarised material.

Student Signature: Zhipeng He Date: October 31, 2021

Acknowledgements

Throughout the writing of this thesis I have received a great deal of support and assistance. First and foremost I am extremely grateful to my supervisors, Dr. Chun Ouyang and Dr. Catarina Moreira for their invaluable advice, continuous support, and patience during my Honours study. Their immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life. I would like to thank all the members in School of Information Systems. It is their kind help and support that have made my studies and life at QUT a wonderful time. Finally, I would like to express my gratitude to my parents. Without their tremendous understanding and encouragement, it would be impossible for me to complete my studies.

1 Introduction

As a new discipline of process mining, predictive process analytics focus on analysing historical data to predict future observations of a business process. These predictions mainly involve the next-event forecasting [7], outcome of an on-going case [25], and the remaining time for a case till its potential completion [30].

In the last decade, a variety of techniques have been used to conduct these predictions. For instance, some methods in the literature use process-specific techniques [16, 26]. while other works use data science techniques and machine learning algorithms [13, 17, 29]. More recently, deep learning techniques have been brought into attention in predictive process analytics due to their high performance in making accurate predictions in text mining and image processing [12]. Similarly to natural language processing that uses sequential algorithms, process analytics takes sequential data as input. Hence, deep learning-based approaches are also applicable to process analytics. As Rama-Maneiro et al. [20] stated in their research, "deep learning has been widely applied to the predictive monitoring of business processes".

Predicting the next event is a typical process prediction problem and an important and challenging topic in predictive process analytics [23]. By applying next event prediction iteratively and progressively, it is possible to obtain a sequence of future events — prediction of remaining sequence. This will ultimately lead to process completion resulting in process outcome prediction. The accuracy of the latter two predictions is depending on the quality of next activity prediction. In addition, time prediction is also a classic regression problem in business predictive analytics. Thus, this thesis intends to determine the extent to next event and time prediction. Also, considering the capability and performance of deep learning, it will serve as the main techniques to underpin the process prediction models in my research.

The overall structure of the study takes the form of six sections, including this introductory section. Section 2 begins by laying out the theoretical dimensions of the research, and looks at the state-of-the-art of business predictive analytics. Section 3 is concerned with the methodology and the experiment design for this thesis. Section 4 presents and analyses the findings from benchmark. Section 5 will discuss the limitation and challenges facing in the research. Finally, Section 6 gives a brief summary and critique of the findings, and includes a discussion of the implication of the findings to future research into this area.

2 Literature Review

This literature review will cover core techniques and knowledge for business process prediction underpinned by deep learning techniques.

2.1 Predictive Process Analytics

Predictive process analytics is an application of predictive analytics in the field of Business Process Management, which predicts the future states of a running business process. The workflow for conducting predictive process analytics is discussed in some papers. Maggi et al. [14] presents a novel framework called *Predictive Business Process Monitoring Framework*. It uses *Trace Processor* module to construct prediction models based on historical event logs and takes the output of previous module as *Predictor* module to process the prediction of current execution trace. This workflow provides a basic idea of how to perform predictive process monitoring, but defines ad-hoc checkpoints specific to certain algorithms and predictions, which makes it hard to apply the approach to other projects.

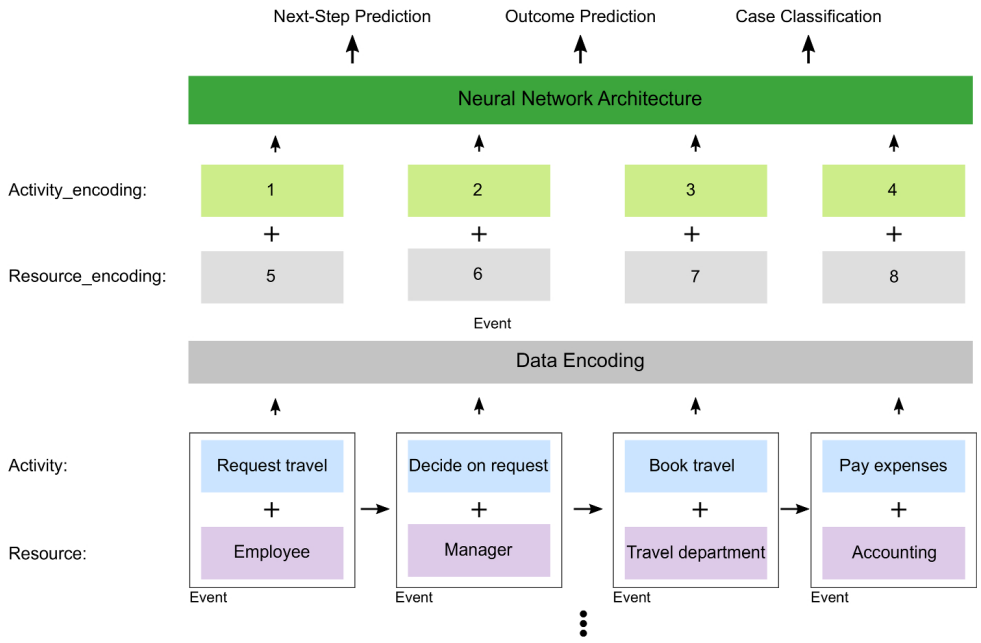

Figure 1: Deep learning based predictive process analytics workflow [18].

To solve this problem, a general approach for predicting business process are proposed by Márquez-Chamorro et al. [15], which divides the workflow into two stages. The first stage runs offline and contains three main checkpoints, such as encoding event logs, building predictive models and evaluating the models. The second stage is online, in which real

time process events are fed to the predictive models and final prediction are generated as the output. This workflow indicates that the most significant step for predictive process analytics is the construction of predictive models because the accuracy of the final results seem to depend on the quality of the predictive model. A more specific workflow based on deep learning approaches as shown in Fig. 1 is introduced by Neu et al. [18], which builds the offline training model by deep learning. It indicated that there are three major components in deep learning-based business process prediction, which are input logs, encode methods and deep learning architectures.

2.2 Event Logs

Event log, as the standard data format for process mining [27], is the main input data for predictive process analytics. It describes the interchanges of on-going process events between information systems or applications. A fragment of real-world event logs from Volvo IT incident management system[1] is shown in Table 1.

Case ID	Activity	Resource	Timestamps	org:group
1-364285768	Accepted + In Progress	Frederic	01.04.2010 0:59:42	V30
1-364285768	Accepted + In Progress	Frederic	01.04.2010 1:00:56	V30
1-364285768	Queued + Awaiting Assignment	Frederic	01.04.2010 1:45:48	V5 3rd
1-364285768	Accepted + In Progress	Anne Claire	07.04.2010 0:44:07	V5 3rd
...	...	...	...	...
1-364285768	Accepted + Assigned	Sarah	12.04.2012 1:11:25	V5 3rd
1-364285768	Accepted + In Progress	Loic	03.05.2012 19:10:10	V5 3rd
1-364285768	Completed + Resolved	Loic	03.05.2012 19:10:12	V5 3rd
1-364285768	Completed + Closed	Siebel	11.05.2012 9:26:15	V5 3rd
...	...	...	...	...

Table 1: Event log example

Each row of an event log represents an execution of process event. For a standard event log, it requires at least three elements in each row, which contain a unique identifier for the case of current event (e.g., Case ID), a name or an identifier for the activity of current event (e.g., Activity) and an execution time of current event (e.g., Timestamps). Some extra attributes are also included in the event logs such as resource name or identifier (e.g., Resource) and resource group information (e.g., org:group), which are associated with the execution of the corresponding event. According to the definition of event log format XES [10], "an event that occurs ... before another event that is related to the same trace shall be assumed to have occurred before that other event." It suggests that the events in the same trace or case should be sequentially ordered by timestamps. Thus, predictive process analytics deals with sequential data.

[1] https://doi.org/10.4121/uuid:500573e6-accc-4b0c-9576-aa5468b10cee

2.3 Encoding

The purpose of encoding is to transform the event data to tensors in which deep neural networks can directly learn and process. Two main encoding methods for predicting next activity and resource, such as one-hot and embedding are presented in the following.

One-hot encoding [9] It focuses on transforming the categorical values, such as the names of activity, to the numerical values, while it has better performance than common method like label encoding. The process of label encoding is to replace a variable with categorical value with an integer (usually start from 0), but the integer value of the variable may affect the predictive results since the categorisation in a direct way assumes that the variables with large values are more important than others. Comparing with common word encoding, one-hot method makes use of one $1 \times N$ matrix (binary vector) to represent the numerical value. The size N of binary vector is depending on the number of possible distinct values for the variable and the position of a one in the vector will determine which category the variable belongs to. In practical usage [24], it is applied on the prediction of next activity or resource by mapping the state of each unique activity or resource to a list of ordered binary. Assuming that the total amount of unique activities is k, a vector of k digits, which is ordered by random or specific methods, in binary (one or zero) can be used to represent the event logs.

Embedding It originates from natural language processing [4] in which a word from a vocabulary set is taken as input by a specific neural network, called *Embedding Layer*. Each word is embedded as vectors into the following neural layers, which are the normal hidden layers for learning the potential insights. When inputting a whole sentence to the models, the deep neural networks would translate the sentence from an ordered word list to a sequence of vectors. Following this idea, the embedding can also be adopted to transform process traces to vector arrays as the input for next activity prediction neural networks [7]. The traces or cases can be considered as a sentence and the activity or event as the word. Moreover, the vocabulary set of language in next activity prediction is represented by the set of activities in the process models.

2.4 Deep learning architectures

For predictive process analytics, two popular deep learning-based approaches will be discussed in the following.

Long Short Term Memory (LSTM) As a special type of **Recurrent Neural Network (RNN)**, it is designed with multiple switch gates to avoid the problems faced by RNN [8]. The cells of LSTM increase four switch gates and two horizon paths to pass the previous state to the current state and control hidden layer to update only relevant information to the memory. Evermann et al. [7] firstly introduce a novel LSTM next-element prediction model by adapting the sentence prediction methods in natural language processing. They compare activities in cases with words in sentences and find that this structure also works for predicting process monitoring. Following this structure, all previous activities are encoded by word embedding as the prefix of ongoing cases to predict the suffix (next activities or the remaining events). Tax et al. [24] also try the method of LSTM in a different way. They do not have the embedded dimension of LSTM cells and the number of neural networks for each layer is also reduced from 500 to 100-150. Although the outcomes of two paper do not have too much difference, the latter method has a better performance because of avoiding overfit of prediction. Based on these two papers, Camargo et al. [6] combine them and propose a new approach for pre-processing and post-processing the input and output in the LSTM prediction models, which give higher accuracy results.

Transformer is initially introduced in natural language translation area [28]. The concept of transformer is based on the sequence-to-sequence model, which has encoder, encoder vector (intermediate vector) and decoder as the main part. For the encoder and decoder, they are actually multiple layers of LSTM. The attention layer in transformer is called *Scaled Dot-Product Attention* and its output is a weighted sum of the values to avoid the effects of long-range dependencies. But a single self-attention mechanism cannot handles a whole sequence in multiple ways due to too many features in dataset. The researchers prefer to stack self-attention mechanisms together to let them work in parallel, just like making use of multiple LSTM cells in RNN methods. The *multi-head self-attention mechanisms* in transformer help this method discover all of these dependencies in one sequential dataset.

Philipp et al. [19] and Agarwal et al. [1] contrast the transformer method with LSTM method for making process prediction. They both indicate that transformer performs better than RNN methods especially when dealing with large and complex datasets. While the above studies confirm the possibility of implementing transformer to predict next events, they do not provide a detailed framework or architectures for guiding further practice. Most recently, a novel transformer framework for predictive process analytics is introduced by Bukhsh et al. [5]. This paper also suggests that transformer performs very well in predicting the next activity, but it does not utilise additional event attributes and only use event prefixes as the input, which means that it is unfair to compare the results directly with other methods. A potential way for validating the real performance of transformer is to conduct a benchmark in the same situation.

2.5 Research Gap

This Honours research project **aims to investigate the impact of different event logs on the performance of deep learning-based process prediction models.** This is motivated by the fact that the same deep learning technique may perform quite differently, in terms of the resulting process prediction model accuracy, on various event logs, as revealed by the existing studies. From the literature review, encoding methods and settings of neural networks have been considered as two factors that would impact the prediction performance. The deep learning-based process prediction workflow shown in Fig. 1 indicated that event logs could be another influencing factor besides the above two factors. In addition, when Neu et al. [18] evaluated the prediction accuracy for each event log, they concluded that the specific characteristics in the event logs might influence the performance of the model. Therefore, identifying potential key characteristics of event logs that may impact on prediction performance is essential for improving process prediction models. However, this is yet underexplored and presents an open research challenge. Hence, I propose the following two research questions to address the above research gap:

RQ1: *Which characteristics of an event log may impact on process prediction model performance?*

RQ2: *How the event log characteristics (identified in* ***RQ1****) affect deep learning-based process prediction model performance?*

3 Methods

The study is conducted in the form of a benchmark and concerned with figuring out how the event logs will impact the performance of deep learning-based business process prediction models. The benchmark design follows the workflow in Fig. 1 and the overall pipeline of experiments are shown below.

Figure 2: Experiment pipeline for benchmark.

The first step of the overall benchmark is to profile the event logs and select suitable input features, corresponding to Section 3.1. Section 3.2 will introduce the definition of predictive tasks for experiments. Then, the input features will be encoded from event log to tensors (Section 3.3), which are ready for feeding into neural networks. Section 3.4 will explain how to construct different neural networks, which is also the third stage of pipeline. After finishing the preparation of experiments, as the first three stages in pipeline, the proposed deep learning models will be put into evaluation and get the results of benchmark, which will be discussed later in Section 4.

3.1 Input features

An event log is used to extract features for a process predictive model underpinned by deep learning. Based on the notion of a trace (which comprises a sequence of events of a case, see Section 2.2), a prefix trace of length l contains the first l events of a trace. As such, multiple prefix traces of different lengths of a case capture the case execution progressively, and are an important input to train a deep learning model for process prediction. Their key concepts are defined as follows:

Definition 1 (Event and attribute [22]) Let $\mathcal{C}$ be the set of case identifiers, $\mathcal{A}$ the set of activity names, $\mathcal{R}$ the set of resource identifiers, and $\mathcal{T}$ the set of timestamps. $\mathcal{E}$ is the set of *events*, and each event has the above *attributes*[2]. For any $e \in \mathcal{E}$: $c_e \in \mathcal{C}$ is the case identifier of e, $a_e \in \mathcal{A}$ is the activity name of e, $r_e \in \mathcal{R}$ is the resource identifier of e, and $t_e \in \mathcal{T}$ is the timestamp of e. If an attribute is missing, a $\perp$ value is returned, e.g., $r_e = \perp$ means that no resource is associated with event e. □

[2]An event can have more attributes but these are not considered in this research.

Definition 2 (Event Log [22]) An *event log* $\mathcal{L} \subseteq \mathcal{E}$ is a set of events. □

Definition 3 (Trace [27]) A *trace* $\sigma \in \mathcal{L}^*$ is a finite sequence of unique events in $\mathcal{L}$. Let $n = |\sigma|$ and $\sigma = [e_1, ..., e_n]$ (where positive integers $1, ..., n$ can be referred to as event index numbers). For all $i, j \in \{1, ..., n\}$: $c_{e_i} = c_{e_j}$ (i.e., all events in a trace refer to the same case). For $1 \leq i < j \leq n$: $e_i \neq e_j$ (i.e., each event appears only once), and $t_{e_i} \leq t_{e_j}$ (i.e., the ordering of events in a trace should respect their timestamps)[3]. □

Definition 4 (Prefix trace [25]) Given a trace $\sigma = [e_1, ..., e_n]$ and an integer $1 \leq l \leq n$, $prefix(\sigma, l) = [e_1, ..., e_l]$ is a *prefix trace* of σ of length l (i.e., it contains the first l events of σ). □

Based on the above definitions, an event e_i in a prefix trace can be represented as a tuple $(a_{e_i}, r_{e_i}, t_{e_i})$, or (a_i, r_i, t_i) as a simplified notation. The time feature is specified to capture the time elapsed from the start event e_1 to the current event e_i of a trace. We assume that each event has a timestamp associated with the completion of the event, and hence the time feature for event e_i is computed as $\Delta t_i = t_i - t_1$. Given a prefix trace $[e_1, ..., e_l]$, three feature vectors can be extracted as input to a deep learning model, which are activity vector $(a_1, ..., a_l)$, resource vector $(r_1, ..., r_l)$, and a time interval vector $(\Delta t_1, ..., \Delta t_l)$.

3.2 Prediction tasks

Business process prediction aims at forecasting the state of next event or remaining sequences until the end of case by a certain event prefix. Three major predictive tasks, including next activity prediction, next resource prediction and next elapsed time prediction will be conducted in the benchmark. Given a event prefix such as $prefix(\sigma, l) = [e_1, ..., e_l]$, e_{l+1} is the next predicted event by a function Ω. For each Ω, it represents a neural network architecture in this thesis.

Definition 5 (Next activity prediction) The next activity prediction problem can be defined as $\Omega_a = a_{e_{l+1}}$. □

Definition 6 (Next resource prediction) The next resource prediction problem can be defined as $\Omega_r = r_{e_{l+1}}$. □

Definition 7 (Interval time prediction) The interval time prediction problem can be defined as $\Omega_t = t_{e_{l+1}} - t_{e_1} = \Delta t_{l+1}$. □

[3]Event index numbers take precedence over timestamps where two events occur concurrently.

3.3 Data encoding

Definition 8 (One-hot encode) The set of one-hot vector $\mathcal{V}$ can be defined as:

$$\left\{ v \in \{0,1\}^n : \sum_{i=1}^{n} v_i = 1 \right\}$$

where n is the total number of unique variable for a feature. □

Definition 9 (Embedding encode) A matrix $W \in \mathcal{R}^{n \times f}$ can be used to describe the embedding shape in each predictive model. The size of the embedding matrix should be the product of the number of unique variable n in a feature and the amount of input channel size f for hidden neural networks in prediction models. □

3.4 LSTM architectures

Due to practical constraints, this thesis cannot provide a comprehensive experiment on all deep learning approaches for business process prediction. This benchmark will adopt three LSTM architectures from Camargo et al. [6] for evaluation in Fig. 3, which consist on a specialized architecture, shared categorical architecture and full shared architecture.

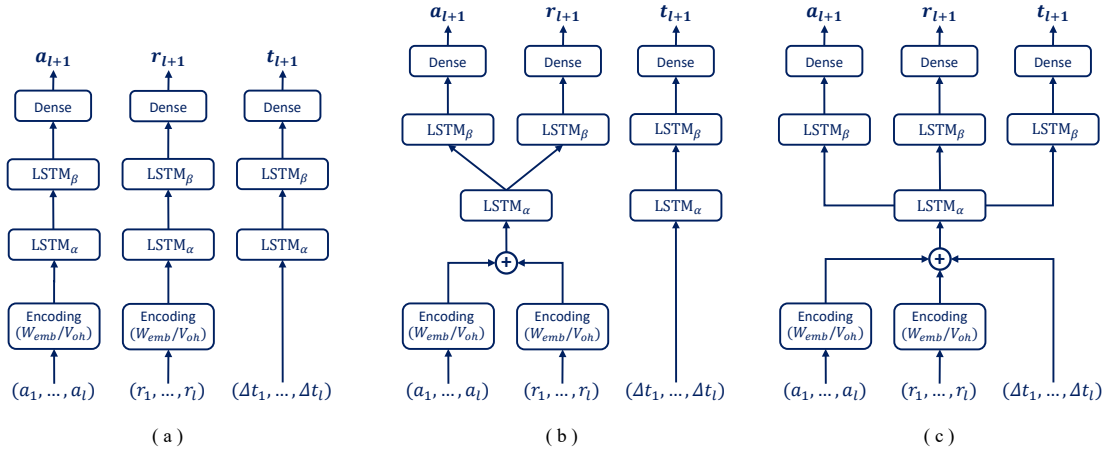

Figure 3: Model architectures of (a) *specialized model*, (b) *shared categorical model* and (c) *full shared model* (adopted from Camargo et al. [6])

All three models share similar settings in the neural networks. They receive the trace prefix $\sigma = [e_1, ...e_l]$ with three fundamental feature vectors as the input of neural networks. For categorical features, activity vector $(a_1, ..., a_l)$ and resource vector $(r_1, ..., r_l)$ are required to go through the additional encoding layer for converting the event data to tensors in

neural networks. While the numerical feature can be feed to networks. Predictive function Ω is represented by the combination of $LSTM_\alpha$, $LSTM_\beta$ and $Dense$ layers, in which $LSTM_\alpha$ are used for learning features and the predictive tasks are done by $LSTM_\beta$. Three predictive task, including next activity a_{l+1}, next resource r_{l+1} and next interval time t_{l+1}, are corresponding to three outputs of the models.

Three LSTM architectures differ in whether sharing information across features:

- In *specialized architecture* (Fig. 3(a)), three features can be recognised as three independent models and they do not share information with others.
- In *shared categorical architecture* (Fig. 3(b)), the categorical features are concatenated into one vector and feed into the same $LSTM_\alpha$. The time feature is still independent from the other two features.
- In *full shared architecture* (Fig. 3(c)), the model will concatenate activity feature, resource feature and time feature into one vector and they will share one $LSTM_\alpha$ layer.

4 Evaluation

4.1 Datasets

For the benchmark, experiments are performed using seven real-life event logs of BPI Challenges concerned with a variety of business processes from different domains:

- BPIC2011[4] event log records the treatment process of Gynaecology department in a hospital.
- BPIC2012[5] event log describes a personal loan application process from a Dutch financial institute.
- BPIC2013[6] dataset is extracted from Volvo's IT incident and problem management system. In this benchmark, only the incident related log will be chosen for experiments.
- BPIC2015[7] dataset contains five event logs on building permit applications from five Dutch municipalities. Since all five event logs share the same application processes but in different municipalities, the experiment will take one of them (BPIC2015-1) into consideration.
- BPIC2017[8] event log is provided by the same loan application process and institute. Comparing with BPIC2012, the new log includes richer cases and information.
- BPIC2018[9] event log records the processes of applications for EU direct payments for German farmers.
- BPIC2020[10] dataset contains a set of event logs from the reimbursement process at TU/e. The benchmark will only use permit log in experiment.

Table 2 provides an overview of the seven event logs containing statistics of the control-flow perspective, such as the number of cases, the number of unique activity and resource, the average and maximum value of trace length and trace duration and the total number of variants.

[4] https://doi.org/10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54
[5] https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f
[6] https://doi.org/10.4121/uuid:500573e6-accc-4b0c-9576-aa5468b10cee
[7] https://doi.org/10.4121/uuid:a0addfda-2044-4541-a450-fdcc9fe16d17
[8] https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b
[9] https://doi.org/10.4121/uuid:3301445f-95e8-4ff0-98a4-901f1f204972
[10] https://doi.org/10.4121/uuid:52fb97d4-4588-43c9-9d04-3604d4613b51

Event Log	Num. cases	Num. activities	Num. Resource	Num. event	Avg. case length	Max. case length	Avg. case duration	Max. case duration	Variants
BPIC2011	1143	624	43	150291	131.49	1814	386.65 days	1156 days	981
BPIC2012	13087	36	69	262200	20.04	175	8.62 days	137.22 days	4366
BPIC2013i	7554	13	1440	65533	8.68	123	12.08 days	771.35 days	2278
BPIC2015-1	1199	398	23	52217	43.55	101	95.72 days	1486 days	1170
BPIC2017	31509	26	149	1160405	36.82	177	21.9 days	281.04 days	15484
BPIC2018	43809	170	165	2514266	57.39	2973	335.39 days	1011.4 days	28923
BPIC2020permit	7065	51	2	86581	12.25	90	87.4 days	1190 days	1478

Table 2: Data profiles for event logs used in experiments.

4.2 Experiment settings

Data split During the preprocessing stage, all event logs have been sorted in chronological order. Each event log will be split into train-test set with a case distribution of 70%:30%. Additionally, 15% of the data from the training set is used as validation split to avoid the overfitting or underfitting in the learning phrase.

Evaluation metrics Determined by different prediction targets, the experiments apply the following metrics for evaluation:

- *Accuracy:* Since the next activity and next resource prediction are both classification problems, the benchmark will utilize accuracy metric. It represents the proportion of all correct classifications in all prediction.
- *Mean Absolute Error (MAE):* While the time prediction task is belong to regression problem, the metric for evaluating next lapse of time is Mean Absolute Error (MAE), which is defined as the arithmetic mean of the prediction errors. Formally,

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

  where y_i is the true value of the test case, $\hat{y}_i$ is the predictive value and n is the total number of test cases.

Feature Selection and Prefix Generation An event log comprises of dynamic and static features. Whilst Dynamic features change over the execution of the process, Static features are often case specific and remains constant. Since all seven logs are made up with several dissimilar features, it is not reasonable to select some unique features into experiments. Based on the experiment design in Section 3.1, only some fundamental dynamic features (shown in Table. 3) will be used for the experiments. In order to predict the activity, resource and elapsed time of next event, it is required to convert them into process prefixes. The method of generating prefix for event log is:

Feature Category	Column Name
Single Process Trace Identification	Case_ID
Dynamic Features	
Activity	Concept_Name
Resource	Resource
Timestamp	Elapsed_Time
Static Features	None

Table 3: Event log features used in the experiments.

- mapping event log by unique Case ID into groups firstly;
- ordering all the dynamic features by 'Timestamp' field in ascending order to generate a process trace for a given Case ID;
- introducing a new time related feature at this stage 'Time Elapsed', which represents the time difference between the very first timestamp of a trace, to the timestamp of a given activity in the same trace;
- generating prefixes at each event, by considering the partial trace from the start of the trace to the given event, eventually.

Implementation Details The experiments were performed on a server with Windows 10 Operation System and its hardware contained 3.8 GHz AMD Ryzen 3900X CPU having 64 GB RAM and one single NVIDIA RTX A4000 GPU with 16 GB Memory. Here are some key ideas for implementation:

- Prefix generation was performed with using Structured Query Language (SQL) by PostgreSQL for speeding up the computation.
- The deep neural networks are implemented by TensorFlow 2.5 Library in Python. The constructed models are trained by an ADAM optimiser with a learning rate of 0.001 for all event logs in the experiments. By default, the maximum number of epoch is set to 200 with batch size of 128.
- Some techniques, like early stopping and adaptive learning rate, are also applied to circumvent the overfitting and underfitting of deep learning models. Generally, the hyper-parameters are keeping consistency for each log and model when training.

4.3 Results and Observations

4.3.1 Next activity & resource prediction

Table 4 and Table 5 present the model performance difference of specialised model, shared categorical model and full shared model for predicting next activity and next resource of seven datasets by accuracy.

	Specialized		Shared categorical		Full Shared	
Dataset	One-hot	Embedding	One-hot	Embedding	One-hot	Embedding
BPIC2011	9.91%	9.91%	9.79%	10.00%	9.91%	9.91%
BPIC2012	78.13%	76.30%	77.38%	77.93%	69.63%	65.97%
BPIC2013i	61.56%	65.01%	60.63%	63.30%	56.15%	59.18%
BPIC2015-1	48.35%	43.77%	45.77%	43.35%	24.20%	33.90%
BPIC2017	63.93%	55.23%	56.02%	53.48%	50.52%	52.81%
BPIC2018	5.38%	5.46%	5.37%	5.45%	5.37%	5.37%
BPIC2020permit	52.48%	52.48%	52.56%	53.52%	30.32%	34.73%

Table 4: The performance comparisons of *specialised model*, *shared categorical model* and *full shared model* in predicting next activity. The measurement metric is accuracy in %.

	Specialized		Shared categorical		Full Shared	
Dataset	One-hot	Embedding	One-hot	Embedding	One-hot	Embedding
BPIC2011	56.18%	56.31%	56.02%	56.27%	56.23%	56.31%
BPIC2012	45.65%	42.07%	63.30%	67.93%	61.97%	65.26%
BPIC2013i	6.50%	6.50%	6.50%	6.50%	6.96%	6.56%
BPIC2015-1	88.96%	88.34%	85.92%	88.29%	41.31%	44.19%
BPIC2017	13.41%	12.83%	43.69%	48.93%	12.91%	12.12%
BPIC2018	15.04%	15.04%	15.04%	15.04%	15.04%	15.04%
BPIC2020permit	74.64%	74.55%	84.92%	87.38%	77.01%	77.01%

Table 5: The performance comparisons of *specialised model*, *shared categorical model* and *full shared model* in predicting next resource. The measurement metric is accuracy in %.

BPIC2011 The results of predicting next activity and resource for all three models are in an approximately same level. The encoding method is not the key reason for influencing the performance of predictive models. However, the accuracy of next activity prediction is only 9.91% and it is extremely low than expected. By contrast, next resource prediction task reaches to around 0.56 in accuracy, which is fair for a complex log.

BPIC2012 From the prediction results of BPIC2012, it is suggested this log is not sensitive to the encoding method and the LSTM models will affect the accuracy instead.

When predicting next activity, *specialised model* and *shared categorical model* achieve a very high value in accuracy (about 0.77). Nevertheless, the accuracy in *full shared model* drops to 65%-68% since the negative impact of the time features. On the other hand, the resource prediction does not work very well in *specialised model*, and the concatenations of activity and resource will assist the improvement of performance.

BPIC2013 Incident The next activity task can get more than 60% of accuracy when only consider activity feature and resource feature as input of models. When introducing the time features, the next activity accuracy will drop to less than 60%. Contrastingly, all three models cannot provide an acceptable result in accuracy of next resource.

BPIC2015-1 The high accuracy in predicting next resource, up to 0.88, shows that event log BPIC2015-1 is receptive to resource features. Also, sharing time feature in *full shared model* has negative influence on the accuracy of both next activity and resource.

BPIC2017 Although BPIC2017 shares the same business process with BPIC2012, the prediction results are thoroughly different. Especially for the next resource prediction, the accuracy is only in value of 12% approximately.

BPIC2018 Similar with the results of BPIC2011, BPIC2018 achieves a stable but low accurate in predicting both next activity (5.37%) and next resource (15.04%). A potential reason for this result is that it consists some significant static attributes and they are not involved in this experiment.

BPIC2020 Permit The permit log within BPIC2020 is comparable to BPIC2015-1, which is also outstanding in resource prediction and can get a reasonable result in next activity task. In practical, *shared categorical model* is the best model for predicting next activity, and *full shared model* will lead to low accuracy in next activity from 0.5 to 0.3.

4.3.2 Interval time prediction

Since the activity feature and resource feature in both *specialised model* and *shared categorical model* are not involved in prediction time-related feature, they can actually be treated as the same predictive model, which only considers time feature as the model input for prediction. In this case, the impact of encoding methods can also be ignored in these two models. For *full shared model*, the experiments on one-hot and embedding would be operated as normal.

Table 6 shows the model performance difference of specialised model, shared categorical model and full shared model for predicting next interval time of seven datasets by *Mean Absolute Error (MAE)* in days. It is indicated that *full shared model*, which introduces categorical features for time predictive task, does not surpass the other two models. In fact, the value gaps of MAE in all seven logs among three models is pretty small. Some logs, like BPIC2015-1 and BPIC2017, get the slight improvement in the prediction after applying *full shared model*. On the other hand, the results of other logs are not benefitted from the change of model structure.

	Specialized & Shared Categorical	Full Shared	
Dataset	One-hot & Embedding	One-hot	Embedding
BPIC2011	268.0638	268.0354	268.1196
BPIC2012	3.527917	4.46625	5.236667
BPIC2013i	46.20542	50.6425	40.4875
BPIC2015-1	34.0875	33.02292	28.58875
BPIC2017	4.991667	5.720833	5.377083
BPIC2018	156.6446	147.9421	147.5338
BPIC2020permit	25.31833	27.07375	23.77583

Table 6: The performance comparisons of *specialised model*, *shared categorical model* and *full shared model* in predicting next interval time. The measurement metric is *Mean Absolute Error (MAE)* in days.

4.4 Analysis and Findings

4.4.1 The impact of encoding method

Fig. 4 shows the different experiments in seven different logs, and the accuracy of models with one-hot encoding do not fluctuate significantly with reference to the accuracy of models encoded by embedding. Comparing with embedding method, one-hot requires a larger vector when the number of unique activity is increasing. It takes more space in memory than word embedding, which would influence the performance of running the models. However, process models are not same as languages because of the different sizes of unique activity and vocabulary. For example, BPIC2013i has only 28 unique activities, which require less encoded vectors than natural languages. From Table 2, even the event log BPIC2011 with the largest number of unique activity is only 624. In this case, one-hot encoding is still usable for predictive process analytics.

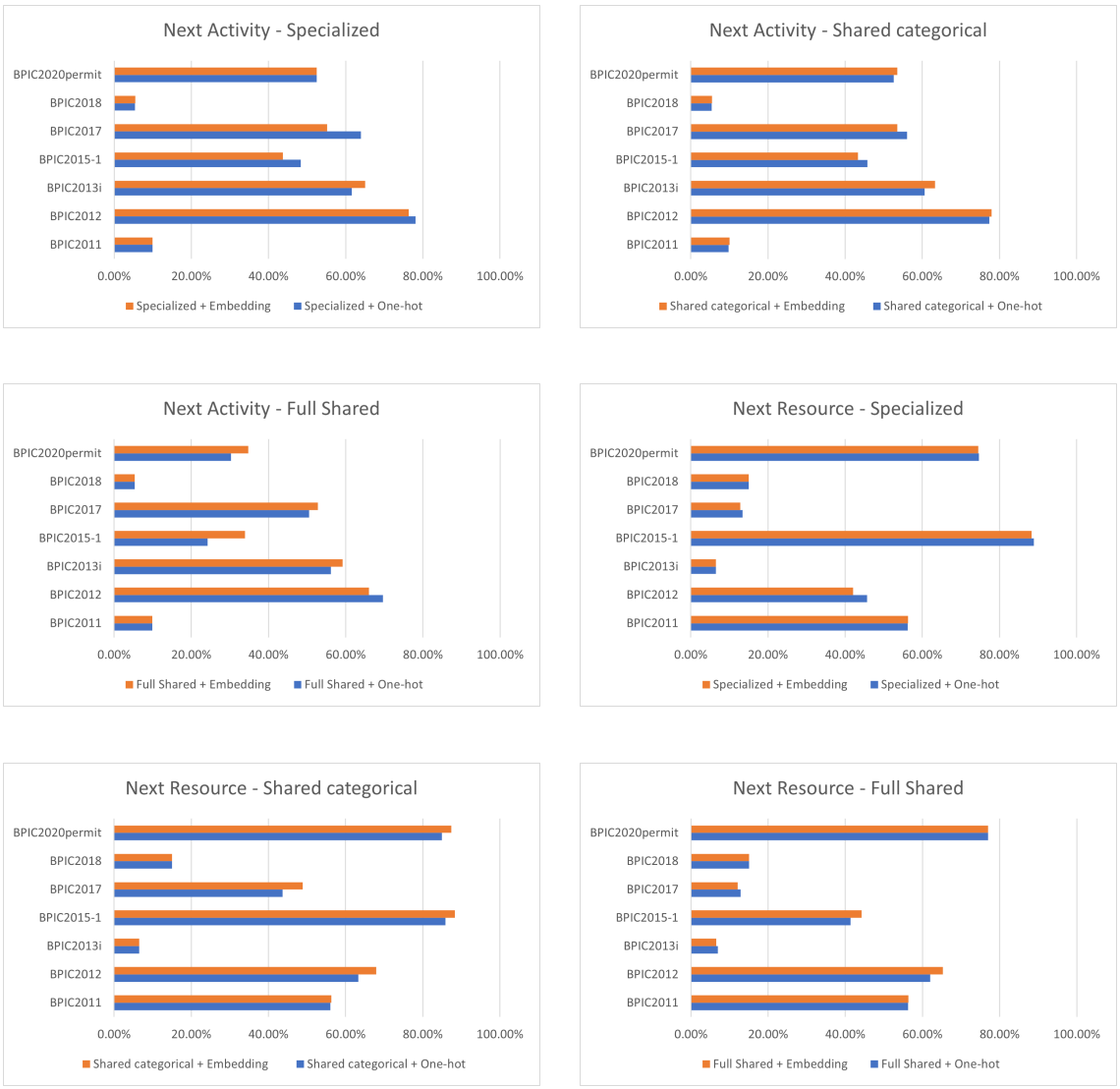

Figure 4: Accuracy differences for One-hot and Embedding in all seven experiment logs.

4.4.2 Activity-weakness and resource-weakness

The benchmark results on all seven logs show that not all event logs are omnipotent in both next activity and resource tasks. Some of them may achieve good performances in computing next activity prediction, while the other logs may not. Based on this characteristic, the event logs can be clustered into activity-weakness logs and resource-weakness logs. The definition of each class will be given below.

- **Activity-weakness log:** Given an event log, if its accuracy of predicting next activity in any of all three deep learning models is less than 33.3%, it can be assumed as

an activity-weakness log. From the benchmark, *BPIC2011*, *BPIC2015-1*, *BPIC2018* and *BPIC2020* permit can be recognised in this category.

- **Resource-weakness log:** Given an event log, if its accuracy of predicting next resource in any of all three deep learning models is less than 33.3%, it can be assumed as a resource-weakness log. The benchmark results suggest that three event logs, *BPIC2013i*, *BPIC2017* and *BPIC2018*, meet this criterion.

Comparing two classes of logs with the statistics of event logs in Table 2, some potential characteristics for these event logs can be surmised as the reasons for low performance in activity prediction. Firstly, the large number of unique activities in an event log would reduce the accuracy in activity prediction. In Fig. 5, it shows the distribution of the number of unique activities for seven logs, in which *BPIC2011*, *BPIC2015-1*, *BPIC2018* and *BPIC2020 permit* has more unique activities than other logs.

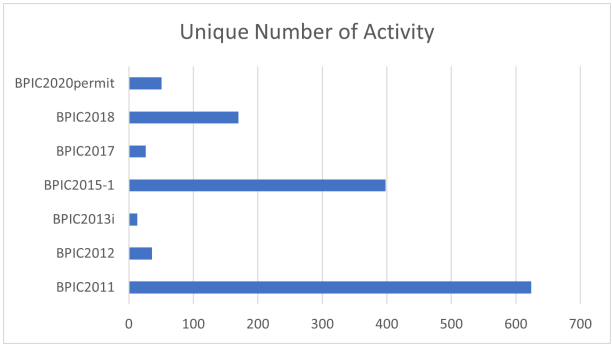

Figure 5: The number of unique activities for seven event logs.

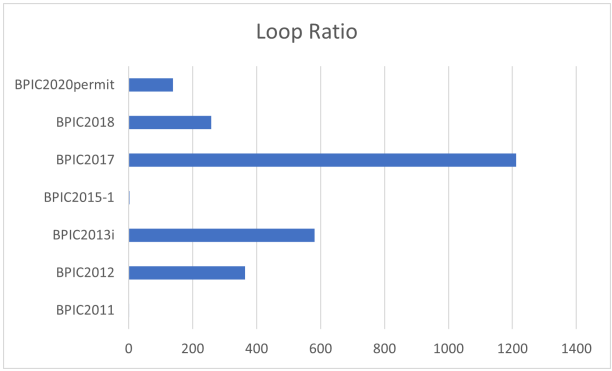

Figure 6: Activity loop ratio for seven event logs.

The degree of iterations and loops in an event log would affect the predictive performance of next activity positively. This measurement of activity loops can be calculated by the total number of events, the number of unique activities and the average length of cases. Assuming that each unique activity repeats the same number of times in one event logs, the ratio of the repeating times of each activity with the average case length can be

recognised as the degree of activity iterations. As the Fig. 6 shown, the activity weakness logs, including *BPIC2011*, *BPIC2015-1*, *BPIC2018* and *BPIC2020 permit*, have low values in loop ratio than other logs with high performance in activity prediction.

On the other hand, the impact factor of resource prediction is similar with the first reason in next activity prediction. The performance of resource prediction (Fig. 7) would reduce when increasing the number of unique resources. Especially, *BPIC2013i* has 1440 unique resources and the accuracy is only 6.5%.

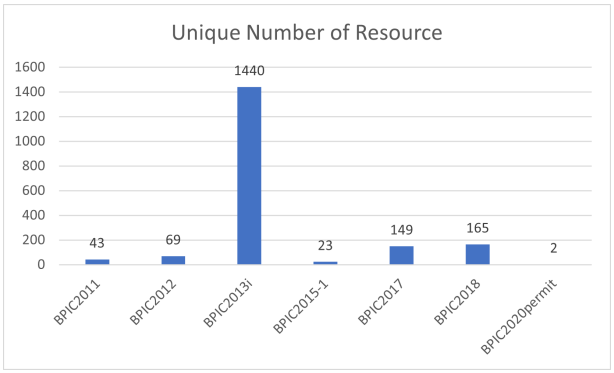

Figure 7: The number of unique resources for seven event logs.

4.4.3 The impact of time feature

From the results of predicting next activity, next resource and next interval time, the time feature plays a limited role in improving the performance. Fig. 8 compares the relationship of the average duration of cases and the result of next interval time prediction, in which the MAE is aligned with the average case duration. It means that the length of the case duration would affect the accuracy of predicting time-related feature.

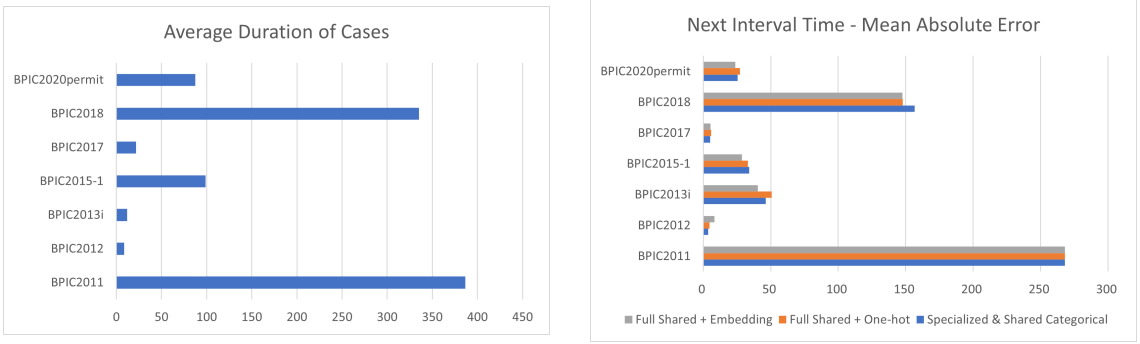

Figure 8: The distribution of average duration of cases and the MAE of next interval time prediction by seven logs.

The time feature will also influence the next activity and resource prediction. It even disturbs the feature learning and leads to the performance reduction in some logs, like BPIC2012 and BPIC2015-1. However, from the analysis of the relationship between the statistics of event logs and the prediction results, no related characteristics can be summarised. Due to practical constraints, this thesis cannot provide a comprehensive explanation of the negative impacts.

5 Discussion

5.1 Clustering prefix traces into buckets

The low accurate predictions in next activity and resource for BPIC2018 indicate that it is not a reasonable choice to put the entire log into models directly, especially for a huge dataset with poor data quality [21]. Normally, from the process analytics perspective, a functional way is to cluster one log by some indicators, which is also known as bucketing. Some researches in machine learning-based predictive process analytics [25, 30] has utilised process bucketing, in which prefix traces can be divided into several buckets and each bucket will be assigned a unique predictor for improving the prediction performance.

Although extensive research in has been carried out on deep learning-based business predictive analytics, no single study exists which considers bucketing as a part of predictive model. A potential reason is that the deep neural networks can learn features by themselves and researchers would like to build more generative models. However, the complexity of event log reduces the performance of neural networks. In particular, some real-life logs, like BPIC2012 [3], record traces from a business process with several sub-processes. It is becoming increasingly difficult to ignore the usage of bucketing with deep learning approaches. Based on the pipeline in Section 3, adding a extra phrase for bucketing the event log before encoding is an appropriate solution, such as split buckets by the prefix length or domain knowledge from logs.

5.2 Challenges in predictive pipeline efficiency

Preprocessing the event log is one of the significant part in benchmark. While experiments in this thesis only did some essential actions, it still takes amount of time for processing all seven logs. Typically, Pandas and Numpy libraries in Python are most common way to process the event logs in the previous research because of the convenience of the built-in functions, like debugging process. However, the performance of them will drop down dramatically when dealing with large real-life logs. For example, it takes more than 24 hours to generate all trace prefixes for BPIC2018 by Python. Hence, this benchmark applied Structure Query Language for improving the efficiency of processing event logs. Despite the fact that SQL requires extra software and code than Python, it is reasonable to trade convenience for efficiency.

Another possible challenges is that the LSTM models used in this benchmark requires a large amount of the computation costs to build, compile and train neural networks. The three LSTM models in architectures have a slightly difference in number of total LSTM layers. For example, *specialised model* requires three $LSTM_\alpha$ layers and three $LSTM_\beta$ layers, and *full shared model* only requires one $LSTM_\alpha$ layer and three $LSTM_\beta$

layers. Hence, the efficiency of each LSTM model should also be taken into consideration if possible.

6 Conclusions

This thesis is aimed to explore the impact of event log on the performance of deep learning-based business predictive models, and identify the potential key characteristics from event logs. In Honours research, the experiments evaluated three LSTM architectures with different structures on seven real-life event logs from BPI Challenge. By analysing the results from predicting next activity, next resource and next interval time, it can be concluded that the characteristics of event logs have both positive and negative affects on the performance of predictive models. Due to the limitation from the time and computation cost in Honours research, the research was limited to some event log characteristics from the statistics of logs. Some further research is needed for discover and investigate more key characteristics in event logs.

The further work should seek to decouple the complex real-life event logs into subgroups by domain knowledge or techniques from data science, such as bucketing. It will benefit for finding more realistic and accurate characteristics in event logs. In addition, the benchmark in this thesis was limited to three LSTM-based models and some up-to-date architectures were not considered in the experiment. It is worth to expand the scope of deep learning models in predicting business processes. For example, the impact of attention layers has received considerable critical attention in researches [11, 31]. Also, besides the RNN-based model, other deep learning techniques, like Transformer [5] and convolutional neural networks [2], can take into consideration. Ultimately, based on the experiments on enough event logs and predictive models, a business process predictive recommendation system can be created for providing information to select suitable bucketing methods, encoding methods and predictive models by analysing the provided event log.

References

[1] P. Agarwal, D. Swarup, S. Prasannakumar, S. Dechu, and M. Gupta. Unsupervised Contextual State Representation for Improved Business Process Models. In *Business Process Management Workshops*, pages 142–154, 2020.

[2] A. Al-Jebrni, H. Cai, and L. Jiang. Predicting the Next Process Event Using Convolutional Neural Networks. In *2018 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pages 332–338, 2018.

[3] A. D. Bautista, L. Wangikar, and S. M. K. Akbar. Process Mining-Driven Optimization of a Consumer Loan Approvals Process. *BPI Challenge*, 2012.

[4] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003.

[5] Z. A. Bukhsh, A. Saeed, and R. M. Dijkman. Processtransformer: Predictive business process monitoring with transformer network, 2021. arXiv:2104.00721.

[6] M. Camargo, M. Dumas, and O. González-Rojas. Learning Accurate LSTM Models of Business Processes. In *Business Process Management*, pages 286–302, 2019.

[7] J. Evermann, J.-R. Rehse, and P. Fettke. Predicting Process Behaviour Using Deep Learning. *Decis. Support Syst.*, 100:129–140, 2017.

[8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[9] J. T. Hancock and T. M. Khoshgoftaar. Survey on Categorical Data for Neural Networks. *J. Big Data*, 7(1):1–41, 2020.

[10] IEEE Computational Intelligence Society. IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams. *IEEE Std 1849-2016*, pages 1–50, 2016.

[11] A. Jalayer, M. Kahani, A. Beheshti, A. Pourmasoumi, and H. R. Motahari-Nezhad. Attention Mechanism in Predictive Business Process Monitoring. In *2020 IEEE 24th International Enterprise Distributed Object Computing Conference*, pages 181–186, 2020.

[12] Y. LeCun, Y. Bengio, and G. Hinton. Deep Learning. *Nature*, 521(7553):436–444, 2015.

[13] F. M. Maggi, C. Di Francescomarino, M. Dumas, and C. Ghidini. Predictive Monitoring of Business Processes. In *Advanced Information Systems Engineering*, pages 457–472, 2014.

[14] F. M. Maggi, C. D. Francescomarino, M. Dumas, and C. Ghidini. Predictive Monitoring of Business Processes. In *CAiSE 2014*, volume 8484 of *Lecture Notes in Computer Science*, pages 457–472, 2014.

[15] A. E. Márquez-Chamorro, M. Resinas, and A. Ruiz-Cortés. Predictive Monitoring of Business Processes: A Survey. *IEEE Trans. Serv. Comput.*, 11(6):962–977, 2018.

[16] A. Metzger, R. Franklin, and Y. Engel. Predictive Monitoring of Heterogeneous Service-Oriented Business Networks: The Transport and Logistics Case. In *2012 Annual SRII Global Conference*, pages 313–322, 2012.

[17] A. E. Márquez-Chamorro, M. Resinas, and A. Ruiz-Cortés. Predictive Monitoring of Business Processes: A Survey. *IEEE Transactions on Services Computing*, 11(6): 962–977, 2018.

[18] D. A. Neu, J. Lahann, and P. Fettke. A Systematic Literature Review on State-of-the-art Deep Learning Methods for Process Prediction. *Artif. Intell. Rev.*, pages 1–27, 2021.

[19] P. Philipp, R. Jacob, S. Robert, and J. Beyerer. Predictive Analysis of Business Processes Using Neural Networks with Attention Mechanism. In *2020 International Conference on Artificial Intelligence in Information and Communication*, pages 225–230, 2020.

[20] E. Rama-Maneiro, J. C. Vidal, and M. Lama. Deep Learning for Predictive Business Process Monitoring: Review and Benchmark, 2021. arXiv:2009.13251.

[21] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. M. Aroyo. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In *2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.

[22] S. Suriadi, C. Ouyang, W. M. P. van der Aalst, and A. H. M. ter Hofstede. Event Interval Analysis: Why Do Processes Take Time? *Decis. Support Syst.*, 79:77–98, 2015.

[23] B. A. Tama and M. Comuzzi. An Empirical Comparison of Classification Techniques for Next Event Prediction Using Business Process Event Logs. *Expert Syst. Appl.*, 129:233–245, 2019.

[24] N. Tax, I. Verenich, M. La Rosa, and M. Dumas. Predictive Business Process Monitoring with LSTM Neural Networks. In *Advanced Information Systems Engineering*, pages 477–492, 2017.

[25] I. Teinemaa, M. Dumas, M. L. Rosa, and F. M. Maggi. Outcome-Oriented Predictive Process Monitoring: Review and Benchmark. *ACM Trans. Knowl. Discov. Data*, 13 (2):17:1–17:57, 2019.

[26] W. van der Aalst, M. Schonenberg, and M. Song. Time Prediction Based on Process Mining. *Information Systems*, 36(2):450–475, 2011.

[27] W. M. P. van der Aalst. *Process Mining: Data Science in Action, Second Edition*. Springer, 2016.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin. Attention is All You Need. In *NIPS'17*, page 6000–6010, 2017.

[29] I. Verenich, M. Dumas, M. La Rosa, F. M. Maggi, and C. Di Francescomarino. Complex symbolic sequence clustering and multiple classifiers for predictive process monitoring. In *Business Process Management Workshops*, pages 218–229, 2016.

[30] I. Verenich, M. Dumas, M. L. Rosa, F. M. Maggi, and I. Teinemaa. Survey and Cross-Benchmark Comparison of Remaining Time Prediction Methods in Business Process Monitoring. *ACM Trans. Intell. Syst. Technol.*, 10(4):34:1–34:34, 2019.

[31] B. Wickramanayake, Z. He, C. Ouyang, C. Moreira, Y. Xu, and R. Sindhgatta. Building Interpretable Models for Business Process Prediction using Shared and Specialised Attention Mechanisms, 2021. arXiv:2109.01419.