A Dashboard for emulating LSTM-based Predictive Process Monitoring and its Qualitative Evaluation

Fazal, Rehan

Award date:
2021

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science
Process Analytics

A Dashboard for emulating LSTM-based Predictive Process Monitoring and its Qualitative Evaluation

Master Thesis Project Report

Rehan Fazal

Supervisor

dr. ir. D. Fahland

Assessment Committee

dr. ir. D. Fahland
dr. ir. M.R.V. Chaudron
dr. ir. R. Medeiros de Carvalho

November - 2021

Abstract

The increasing popularity of machine learning methods has led to the development of various applications that are designed to improve the efficiency of several business processes. Among these is predictive process monitoring that use event logs to provide forecasts and predictions for various business processes. However, with the growing interest in the research field, abundant work has been done to build an accurate predictive model rather than visualise its outcome for real-world business users, specifically the domain experts, for their interpretations. In this Master's Thesis, we address this gap by developing a Dashboard framework that assists business users in making decisions on a running case by providing multiple recommendations with the confidence the model has, which allows them to choose actions leading to a successful outcome. Furthermore, this thesis also intends to make the business user explain the model predictive recommendation capability and also allow the user to provide feedback to the predictive model using the dashboard's What-If analysis. Finally, we evaluate the dashboard through demonstration with the business understanding extracted from the process model of the event log.

Preface

This master thesis is the result of my graduation project for the Data Science in Engineering master at the Eindhoven University of Technology, conducted within the Process Analytics group at the Department of Mathematics & Computer Science.

To begin, I would want to express my sincere gratitude to Dr Dirk Fahland for giving me this wonderful chance and for his assistance during this whole thesis. I was able to produce the work in front of you today as a result of his hard questions, innovative ideas, and excellent feedback. Thanks for taking the time to listen to my concerns and respond to all of my queries.

Furthermore, I want to express my gratitude to my parents for their unwavering compassion and support throughout my life. Thank you for letting me develop in the manner in which I see fit. I would also want to express my gratitude to my sister for staying with the family through the COVID-19 pandemic and taking care of them. Finally, but certainly not the least, I would want to express my gratitude to all of the friends that I met during my studies.

Contents

Chapter 1

Introduction

In this chapter, we first provide a general context of this thesis in Section 1.1. We then present the objective and related research question in Section 1.2. Lastly, in section 1.3 we outline the method used to perform the research.

1.1 Thesis Context

Process mining is the study of examining process logs to uncover representations of underlying process models, compare process models with event logs, or optimize business processes [1]. Predictive process monitoring (PPM) is a subfield of process mining that focuses on proactively monitoring business processes to predict the unfolding of ongoing processes based on the knowledge from historical event logs [2]. This is performed by the development of a prediction model using historical data, referred to as offline phase in [3], which helps to make predictions of the running case called online phase [3].

In recent times, there has been growing interest in building prediction models by applying machine learning, specifically sequential deep learning techniques within the domain of PPM that are being researched to predict business process responses, such as the next event in a case, the time for completion of an event, and the remaining execution trace of a case [4–6]. Despite their high accuracy and advanced predictive capabilities, these models do not provide much insight into why a prediction was made and the possibility of alternate predictions. It acts as a black box [7, 8]. This results in demand for interpretable predictions so that the user can evaluate the various predictions and their underlying explanation with the business knowledge and with the information presented to them, to take actions accordingly. This thesis focuses on the online phase of the predictive process monitoring dashboard framework for business users who are domain experts. Usually, these users have little expertise in machine learning and have little desire and capacity to comprehend the adopted black-box models. So, they look for an explanation to justify the model's outcome from the context of the situation and their business acumen [9].

However, the applicability of the existing state-of-the-art methods in the online phase is limited as there has been no study conducted on how an online phase dashboard framework for a domain expert should look like and what information would be useful for them to see on the dashboard which could help them to make decisions for an ongoing case. The thesis explores designing a dashboard front-end that can display the predictions generated by a deep learning backend into a list of multiple recommendations where a user can recognize the most appropriate next action for an ongoing case based on explanations provided using contextual information, reliability of the prediction and future insights.

1.2 Research Question

Our research goal is to enable business users such as domain experts to understand the explainability of predictive models. The existing sequential deep learning predictive models (LSTM) within the domain of PPM focus on predicting next events using only activity, time, and role as their feature vectors [4–6], leading to limited accuracy, performance and explainability, which brings to our first research question:

Research Question 1 : *How to improve the prediction of the running case for the selected LSTM base model for the given business process for more explainability?*

Once the predictive model design is improved for more explainability, the primary objective, while solving the problem discussed in Section 1.1, is to not only make predictions about the next event but also to provide alternate recommendations that would allow to still reach a successful outcome. Thus, with each multi-recommendations, information such as past executed actions, time duration, resources involved, contextual information, reliability of prediction and user-defined labels based on which the user can judge which next action is optimal or desired for the business. This also ensures that the recommendations for a specific case can also be focused on predictions in other, similar cases. So, we need to answer these two research questions:

Research Question 2 : *How should the dashboard be designed to incorporate the LSTM models to simulate the predictions generated for the users?*

Research Question 3 : *How the model will provide the alternate recommended actions for an event which could lead to a successful outcome?*

After developing the dashboard intended for business users, the quality of the generated multiple recommendations has to be evaluated among them, which brings us to our next research question:

Research Question 4 : *How does the multi recommendations perform with each other?*

Once the qualitative evaluation on the multi recommendations has been performed, we need to insure how the dashboard works for domain experts to help them make decisions, which brings us to our next question:

Research Question 5 : *How to use, translate, and explain the confidence on the recommendations provided by the model with the available information from the dashboard?*

1.3 Research Method, Outline & Results

We used the Sepsis Cases event log, which is a real-life event log of treatment of patients suffering from life-threatening sepsis symptoms at a Dutch hospital. Then, we discuss the related work in the sequential deep learning models in the PPM space and select the predictive model. This will be followed by outlining the gap in the PPM which might be suitable for the domain expert's use case. With these related work in hand, we will be able to address our research questions listed in Section 1.2 by following these steps:

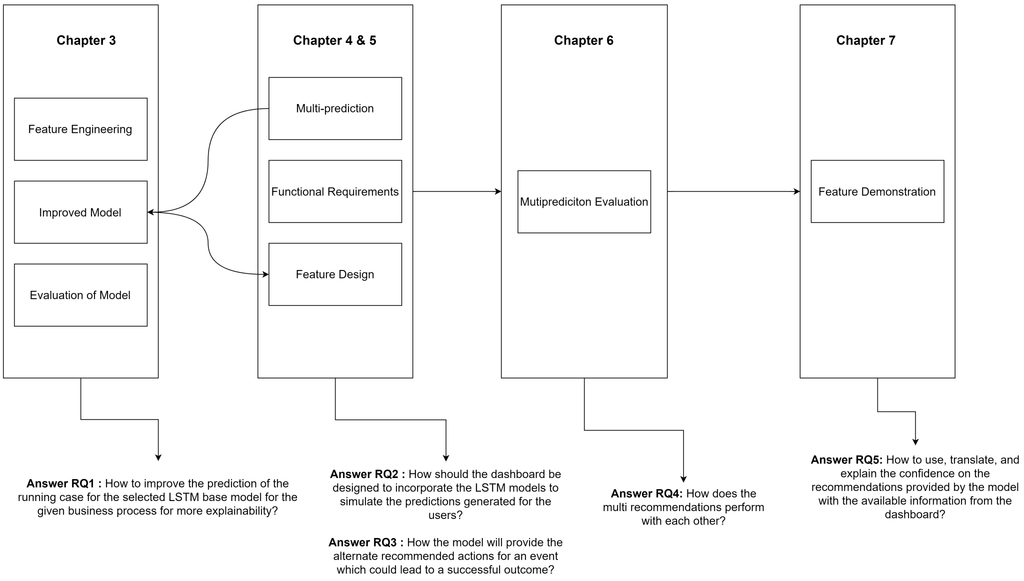

Chapter 3
Feature Engineering
Improved Model
Evaluation of Model
Chapter 4 & 5
Multi-prediction
Functional Requirements
Feature Design
Chapter 6
Mutiprediciton Evaluation
Chapter 7
Feature Demonstration
Answer RQ1 : How to improve the prediction of the running case for the selected LSTM base model for the given business process for more explainability?
Answer RQ2 : How should the dashboard be designed to incorporate the LSTM models to simulate the predictions generated for the users?
Answer RQ3 : How the model will provide the alternate recommended actions for an event which could lead to a successful outcome?
Answer RQ4: How does the multi recommendations perform with each other?
Answer RQ5: How to use, translate, and explain the confidence on the recommendations provided by the model with the available information from the dashboard?

Figure 1.1: Outline Research

1. We improve the selected predictive model explainability and accuracy by choosing meaningful features which explain the actions using a tree-based

classifier from the sepsis event log. Then we encode them in the predictive model and evaluate them quantitively and qualitatively. Qualitative evolution is performed using an industrial process mining tool (Apromore). We then make changes in the predictive model to generate multi predictions with confidence (reliability) associated with each predictive recommendation.

2. Next, we formulate the dashboard's functional requirements, construct a conceptual design of the user interface, and develop them based on the functional requirements. The features required are designed such that it solves the domain expert use case outlined by [9] with the Explainable AI objectives formulated by [10].

3. Two processing mechanism are developed, Single Event processing which will emulate one case at a time using different features it offers and thus explaining the model behaviour using them. Whereas the Batch Processing will be used for evaluation over the test event log.

4. The predictive techniques used in the Single event processing for multi prediction are evaluated using event similarity and log similarity measures with the help of different options offered by the Batch Processing mode. The multi prediction is compared with each other with the objective to see variability in predictive recommendations. The predictive technique associated with replying of historical events offered more variability and similarity, making it an ideal choice for predictive recommendations.

In the end, we demonstrate the Single Event Processing features from the user perspective with a concrete objective for the outcome to be successful or not successful. The interpretation of the recommendations is solely based on the knowledge acquired from the process model of the Sepsis Event log. We categorize the activity in different segments and look out for interchangeability and concurrency. Using the process knowledge acquired, we try to explain each of the mode outcomes. The objective set for the demonstration turns out to be successful for two features and partially successful for one. However, we were able to showcase the explainability objective of each option.

Our methodology is summarised in Figure 1.1 and is marked with the research question concerning the scope of our research. Chapter 2 introduces the necessary background knowledge. In Chapter 3 we improve the selected predictive model. Chapter 4 deals with outlining the functional requirements, designing, and the overview of architecture of the dashboard. Chapter 5 consists of the design of different features introduced in Chapter 4. Chapter 6 consist of the evaluation of predictive techniques designed in chapter 5 over the entire event log. Chapter 7 consist of the demonstration of the dashboard from the user’s perspective. Finally, we conclude in chapter 8.

Chapter 2

Background and Related Work

This chapter gives an overview of the related disciplines and related work that fall within this thesis. Predictive process monitoring is a cross-disciplinary field, it draws the ideas from process mining and machine learning. So, first the process mining research field is introduced with the necessary definitions in Section 2.1. The Subsection 2.2 introduces the neural networks and relevant deep learning techniques. Then in Section 2.3 describes the predictive process mining, which introduces techniques to predict next activities and related work in generating process prediction using sequential deep learning. Lastly, in Section 2.4 we discuss the relevant concerning explainability.

2.1 Process Mining

Process mining is relatively new research discipline which blends the business process management and data mining on one hand but also involves process modelling and analysis. The primary objective of process mining is to aid in the discovery, analysis, redesign, monitoring, and enhancement of business processes using extracted event logs from process execution data [1, 11]. Process mining may be divided into three categories, as illustrated in Figure 2.1: discovery, conformance and enhancement. Their functions are different from one another in analysing event log in the context of processes. In the following subsections, the key process mining concepts utilized in this thesis are introduced.

Event Log

The data which is collected from the execution of a business process is termed event log, and it is the starting point of process mining. Event log records all completed data related to the process. It is composed of cases, each of which represents a distinct occurrence of the business process. Each case is comprised

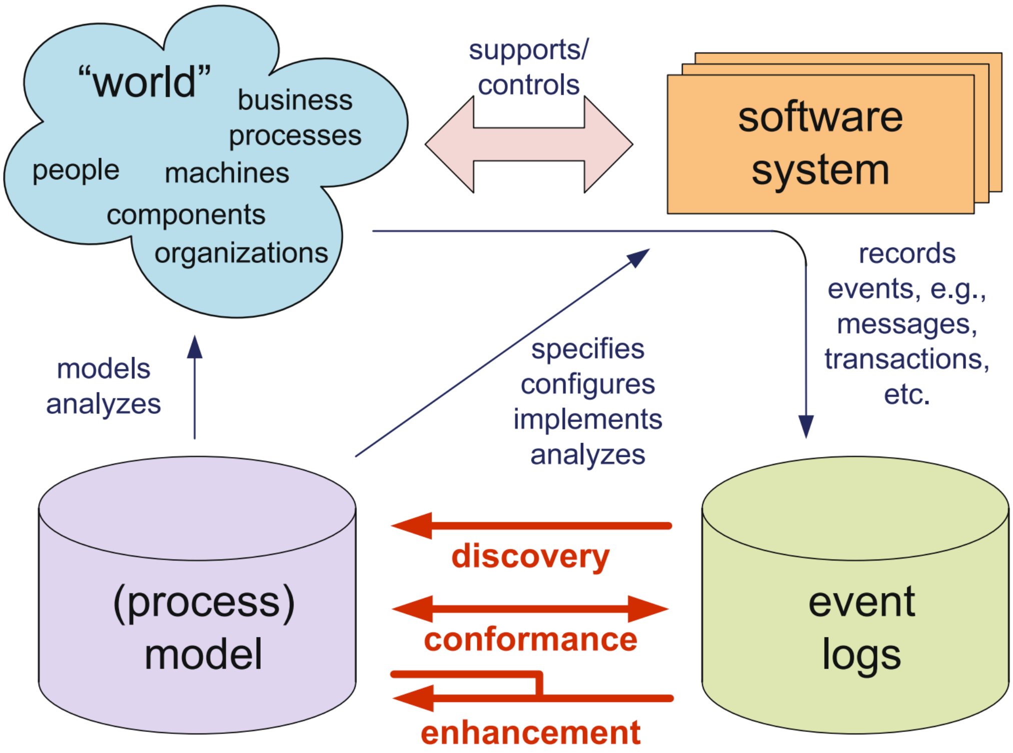


Figure 2.1: Types of Process Mining in relation to one another [12]

of a series of events, each of which represents the execution of a specific process activity. Each event has a variety of attributes, three of which are mandatory:

1. Case Identifier : Indicates the case and process instance resulted in the event.
2. Activity : Identifies the activity to which the event relates.
3. Timestamp : Recordes the moment the event happened.

An event may relate to the start or end of an activity, and the corresponding timestamp will relate to it. In this study, each event log just contains the completion timestamp. Apart from the case identifier, activity, and the timestamp, an event may include other characteristics such as the resource that executes the activity and the additional attributes related to it, and this definition is taken from [13].

Definition 2.1.1 (Events)**.** An *event* is a tuple $(c,a,t,(r_1,v_1),...,(r_m,v_m))$, where c is the case identifier, a is the activity name, t is the timestamp, and $(r_1,v_1)...,(r_m,v_m)$ are event attributes and the corresponding values assumed by them, where $m \geq 0$.

Assume $\mathscr{E}$ be the event universe, i.e., the set of all possible event classes, and $\mathscr{T}$ be the time domain. Then there is a function $\pi_{\mathscr{T}} \in \mathscr{E} \rightarrow \mathscr{T}$ that assigns timestamps to events, and $\pi_{\mathscr{A}} \in \mathscr{E} \to \mathscr{A}$ that assigns activity to each event from the finite set of activities $\mathscr{A}$. The sequence of events generated for the specific case instance forms a trace. Formally,

Definition 2.1.2 (Traces)**.** A *trace* is a non-empty sequence $\sigma =< e_1,...,e_n >$ of events such that $\forall i \in [1..n], e_i \in \mathscr{E}$ and $\forall i, j \in [1..n], e_i \cdot c = e_j \cdot c$. That is to say, that all the events in the trace relate to the same case.

So, a set of completed traces is known as an *event log* $\mathbb{L}$, i.e., $\mathbb{L} = \{\ \sigma_i : \sigma_i \in \mathbb{S}, 1 \leq i \leq \mathbb{K}\}$, where $\mathbb{S}$ is all possible traces and $\mathbb{K}$ is the number of traces in the *event log*.

The prefix events associated with a trace relate to already occurred events, while the suffix events refer to the running case's future course of the trace. Formally,

Definition 2.1.3 (Prefix)**.** For a given trace $\sigma = [e_1,...,e_n]$ and a positive integer $l \leq n$, then $prefix(\sigma, l) = hd^l =< e_1,...,e_l >$.

Definition 2.1.4 (Suffix)**.** Similarly, for a given a trace $\sigma = [e_1,...,e_n]$ and a positive integer $l \leq n$, then $suffix(\sigma, l) = tl^l =< e_{l+1},...,e_n >$.

A fragment of the sepsis event log is presented in Table 2.1 where each case refers to the treatment of one patient. The patient *Age* can be considered as *case attributes*. A case attribute remains constant throughout the given case and is shared by all the events relating to that case. Additionally, *event attributes* may change for each event, for this fragment *Activity, Complete Timestamp, User, Leucocytes* changes from an event to another.

Table 2.1: Extract of sepsis Event Log of a treatment process

Case ID	Event Attributes				Case Attributes
	Activity	Complete Timestamp	User	Leucocytes	Age
CL	ER Registration	06-02-2014 01:17:00	A	0	70
CL	ER Triage	06-02-2014 01:22:09	C	0	70
CL	ER Sepsis Triage	06-02-2014 01:22:30	A	0	70
CL	Leucocytes	06-02-2014 01:26:00	B	11.6	70
OF	ER Registration	09-04-2014 11:43:42	A	0	50
OF	ER Triage	09-04-2014 11:45:43	C	0	50
OF	ER Sepsis Triage	09-04-2014 11:46:15	A	0	50

Conformance Checking

Conformance Checking refers to the process of determining if reality, as recorded in the log, conforms to the model and vice versa [12]. This method takes into account both the model and the event log, and the log is "replayed" on the model. This enables the identification of differences between anticipated and observed behaviour, as recorded in the event loge, and if possible preventive actions can be taken against the undesired behaviour.

In the context of this thesis, the prefixes from the trace will be used to replay over the model, to check the reality of the predictions it is recommending. The model might or might not represent the real business scenario, which will affect the confidence in the recommendations made.

2.2 Neural Networks : RNN and LSTM

A neural network is composed of an input layer, one or more layers of "hidden" cells, and an output layer. All the layer's cells are linked through weighted connections to cells in the preceding and subsequent layers, enabling for different network architectures. The output of each cell is a function of the weighted total sum of its inputs. Gradient-based optimization using training data is used to learn the weights of this weighted sum [14].

Recent developments in the area of predictive business process monitoring have shown the use of deep learning methods [4–6]. Deep learning is a subfield of machine learning that aims to model high-level abstractions of data via the use of networks composed of multiple layers of interconnected neurons with complex structures or non-linear transformations [15]. Through the non-linear transformation, the network is being trained to the patterns and behaviours that can be observed in the data. Theoretically, the combination of complex functions allows higher-level patterns to be detected in data with more layers of *neurons in the network* [16].

RNN

Recurrent Neural Networks (RNN) are a special type of neural networks which is composed of cyclical connections intended for the prediction of sequential inputs. In this type of input (data), the state of an observation depends on the state of its previous ones. The state of an observation is determined by the state of its predecessor in this kind of input (data). Thus, RNNs use a portion of the processed output (h) from the previous cell to process a new input (X) as depicted in the Figure 2.2. While, RNNs do well when it comes to predicting sequences with short-term temporary dependencies, but they fall short when it comes to long-term dependencies because of the gradient of the loss function decays exponentially with time causing the vanishing gradient issue [14]. This issue is addressed by Long Short-Term Memory (LSTM) networks.

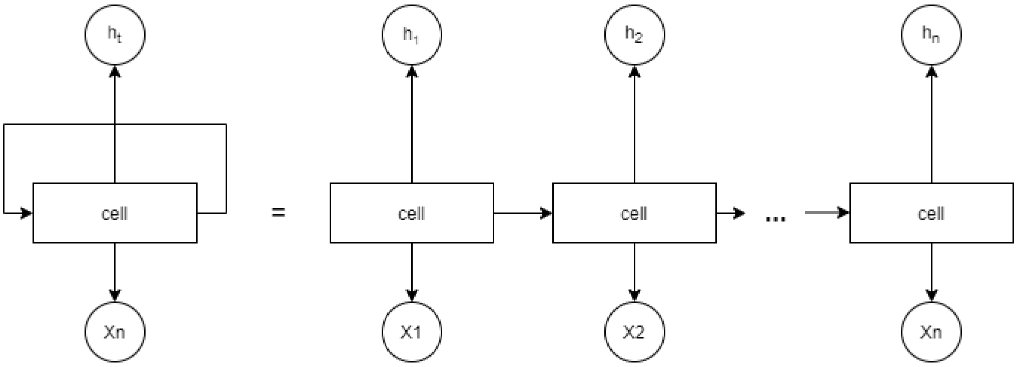


Figure 2.2: RNN basic structure

LSTM

LSTM networks employ in addition to using a portion of the prior output for new processing by implementing a long-term memory using memory cell that can store information over time. As shown in Figure 2.3 an LSTM consists of three gates: an input (i_t), a forget (f_t), and an output (o_t). These gates are referred to as controlling gates because it regulates, when information is captured into memory, when it is forgotten, and when it is processed [17]. The information moves from cell to cell with minimal variation, i.e., only when f_t is activated, previous memory cell status is cleared, which helps to keep some characteristics constant throughout the processing of all inputs. This constant input process enables the predictions to stay meaningful over long durations.

2.3 Predictive Process Monitoring

Provided a completed event log of a business process cases, and a prefix case of this process as received from an event stream, the goal is to predict the performance measure of a prefix case in the future. For instance, predicting the next activity of the given prefix, how it might unfold upon completion or the time required to complete the case. The task is presented in Figure 2.4.

A *prediction point* is the moment in time when the prediction is made, and the predicted value is known by the performance measure. It is usually determined by the amount of data the predictor has available for the prediction. As a result,

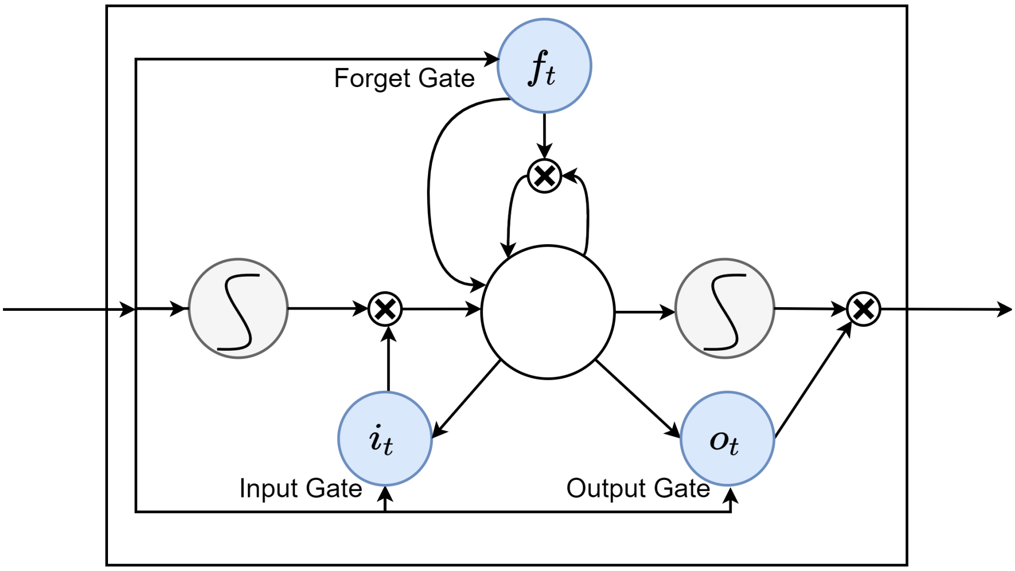


Figure 2.3: LSTM Illustration

a prediction is based on the predictor's knowledge of the process's history up to the prediction point which is supported by the predictor's memory, as well as information of the process's future up to the prediction point which is based on the predictor's ability to perform forecast using trend and seasonal pattern analysis [13].

This approach only requires sufficient amounts of logged case executions to train the predictive model, which is used to make the prediction. In the recent studies, Recurrent neural networks (RNNs) equipped with two hidden layers were discovered to be an ideal fit for predictive model because of the sequential structure of business processes [4–6]. Furthermore, an LSTM technique is endowed with the capacity to take into consideration also certain pre-existing information about the future of an ongoing case [18]. The following subsections entail how such predictive model is achieved and recent advances.

2.3.1 Learning LSTM-based Models for Process Prediction

The formal definitions presented here for ML problem with LSTM-based models, based on [4, 5] work. The learning problem of an activity function is denoted by, f_a^1 and a time prediction function f_t^1 such that $f_a^1(hd^l(\sigma)) = hd^1(tl^l(\pi_{\mathscr{A}}(\sigma)))$ and $f_t^1(hd^l(\sigma)) = hd^1(tl^l(\pi_{\mathscr{T}}(\sigma)))$ for prefix length l. Each event $e \in hd^l(\sigma)$ is transformed into feature vector for LSTM inputs, $x_1, ..., x_l$ either by using on-hot encoding or embeddings. Before doing that, the set of activities $\mathscr{A}$ are converted to $index \in \mathscr{A} \rightarrow \{1, ..., |\mathscr{A}|\}$. For example, in the Table 2.1, it has the following

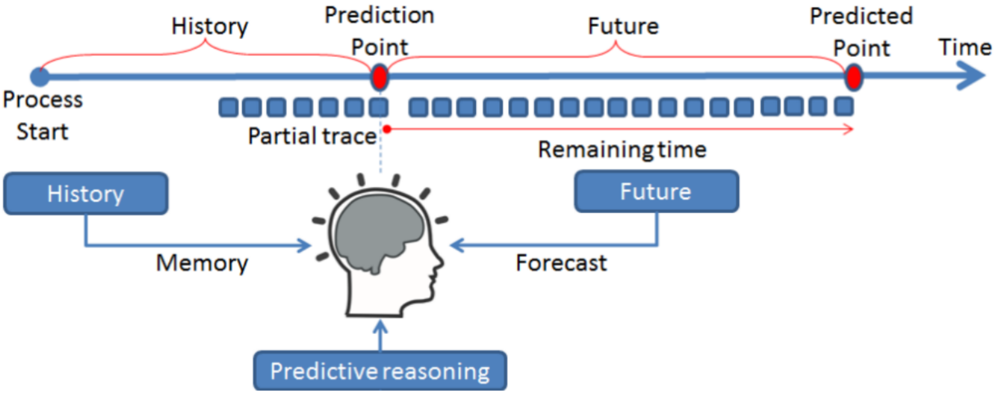


Figure 2.4: Overview of predictive process monitoring, Taken from [13]

set of activities <ER Registration, ER Triage, ER Sepsis Triage, Leucocytes>, each of them will be assigned following index based on the above function ER Registration → 1, ER Triage → 2, ER Sepsis Triage → 3, and Leucocytes → 4.

2.3.1.1 One-hot-encoding

One-hot encoding is typically used for features that have a small to medium number of unique values. Each feature has its own n column representation where $n = |\mathscr{A}|$. The one-hot encoding assigns the value 1 to the feature number $index(\pi_{\mathscr{A}}(e))$ and a value of 0 to the other features. This sort of encoding is also referred to as "dummy encoding".

2.3.1.2 Embeddings

An embedding method is used to learn the multidimensional real valued representation of categorical values. Given the *index* valued representation of activities $index \in \mathscr{A} \to \{1,...,|\mathscr{A}|\}$, it is feed into an extra layer of linear neurons called embedding layers and that maps each index integer value to an entity embedding, which is a fixed size matrix [19]. These embeddings are used, by loading the embedding matrix weights into the respective training network.

2.3.1.3 Ordinal Encoding

Numerical attributes, related to temporal information, do not require encoding since their real value can be processed by the network after the normalization. As a result, the only change undertaken on this numerical data is the padding operation, which produces vectors of fixed size for the varying prefix lengths.

2.3.1.4 Next Event and Suffix Prediction

Using the functions f_a^1, f_t^1 repeatedly, allows predicting the next event until the end of the case. $f_a^{\perp}$, $f_t^{\perp}$ referred to as activity and time until next event functions, which could predict the whole continuation of a running case such that $f_a^{\perp}(hd^l(\sigma)) = tl^l(\pi_{\mathscr{A}}(\sigma))$ and $f_t^{\perp}(hd^l(\sigma)) = tl^l(\pi_{\mathscr{T}}(\sigma))$ and these next event predictions are used repeatedly to make suffix predictions, where $\perp$ represents end of case. Formally complete suffix of activities are calculated as :

$$f_a^{\perp}(\sigma) = \begin{cases} \sigma & \text{if } f_a^1(\sigma) = \perp \\ f_a^{\perp}(\sigma \cdot e), \text{ with } e \in \mathscr{E}, \pi_{\mathscr{A}}(e) = f_a^1(\sigma) \wedge & \\ \pi_{\mathscr{T}}(e) = \left(f_t^1(\sigma) + \pi_{\mathscr{T}}(\sigma(|\sigma|))\right) & \text{otherwise} \end{cases}$$

2.3.2 Recent advances in LSTM-based PPM

This SUBsection discusses recent advances in predictive process monitoring, with a particular focus on LSTM-based next event prediction.

Tax et al. [4] approach predicts the next event in a running case and the time until the next event using an LSTM-based architecture, and these next event predictions are used repeatedly to make suffix predictions. Each event is assigned a feature vector by encoding the categorical event type using one-hot encoding and complementing it with information about the event's occurrence time. The research employs a wide range of architectural designs, as shown in Figure 2.5. The design concerning this research study utilizes a hybrid approach by applying a common LSTM layer to feed two separate LSTM layers : one specialized in predicting the next event and another in predicting time. The study shows that multitask learning approach, i.e., predicting activity and timestamp, performs better than learning each task individually. Although, when the number of event types is low, this approach works well; but, when the number of event types is high, it is adverse, which is addressed in the next study.

Evermann et al. [5] applies LSTM network to just predict the activity and role of the next event of an ongoing trace, although, they do acknowledge that other attributes, such as the event's duration time, may be predicted as well, but the framework developed cannot handle numerical variables, so it could not predict the timestamp. In contrast to [4], this method leverages the embedded dimension of LSTMs to represent categorical attributes by high dimensional continuous vectors and incorporate extra attributes such as the role (resource) associated with each event. It utilizes two hidden layer LSTMs, the dimension of the input changes according to the embedding representation, tenfold cross-validation, and a dropout of 0.2 for each cell.

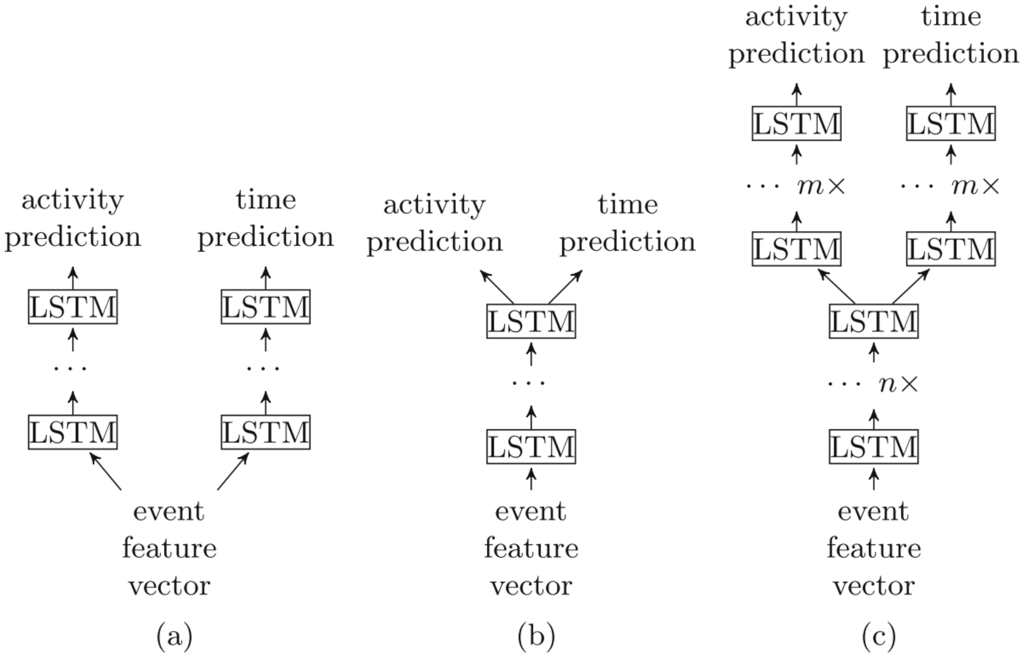


Figure 2.5: Tax et al. LSTM architectures with (a) single-task layers, (b) with shared multitasks layer, and (c) with n + m layers of which n are shared, Taken from [4]

Lin et al. [20] propose an encoder-decoder based framework based on LSTMs, namely MM-Pred, to predict the next activity of an ongoing case. It takes into account on all accessible data points in the input log, namely, the control-flow information (event type) and the case data (event attributes). Each event's attribute is randomly embedded. The encoder converts the input sequence into a set of high-dimensional vectors, which the decoder then converts back into a new sequence for use in the prediction step. This study also doesn't support predictions of numerical attributes, which in-turn no prediction of timestamp just like [5].

Recent study by *Camargo et al.* [6] outlines a method for developing accurate LSTM-based models to predict process behaviour. The work takes into account of using the embedding dimensions from [5]. Along with that it also incorporates the concept of specialized and shared layers, i.e., two LSTM layers, from [4] to design prediction architectures that are capable of handling a large number of different event types with specialized layers for the activity and resource attributes (role) which is similar to multitask learning approach. The method is divided into three phases: pre-processing, which extracts n-grams or fixed-length sequences of events, training using LSTM, and post-processing, which selects the predicted next event. The objective of *Camargo et al.* is to directly include interpretability into an

LSTM-based predictive model by concatenating the information at different points in the network for business process behaviour prediction without compromising accuracy. The study simultaneously predicts the next event and its respective timestamp.

In this thesis, we will incorporate *Camargo et al.* [6] full shared LSTM architecture because of its advantage of predicting activity, role, and time over other exiting LSTM based predictive model.

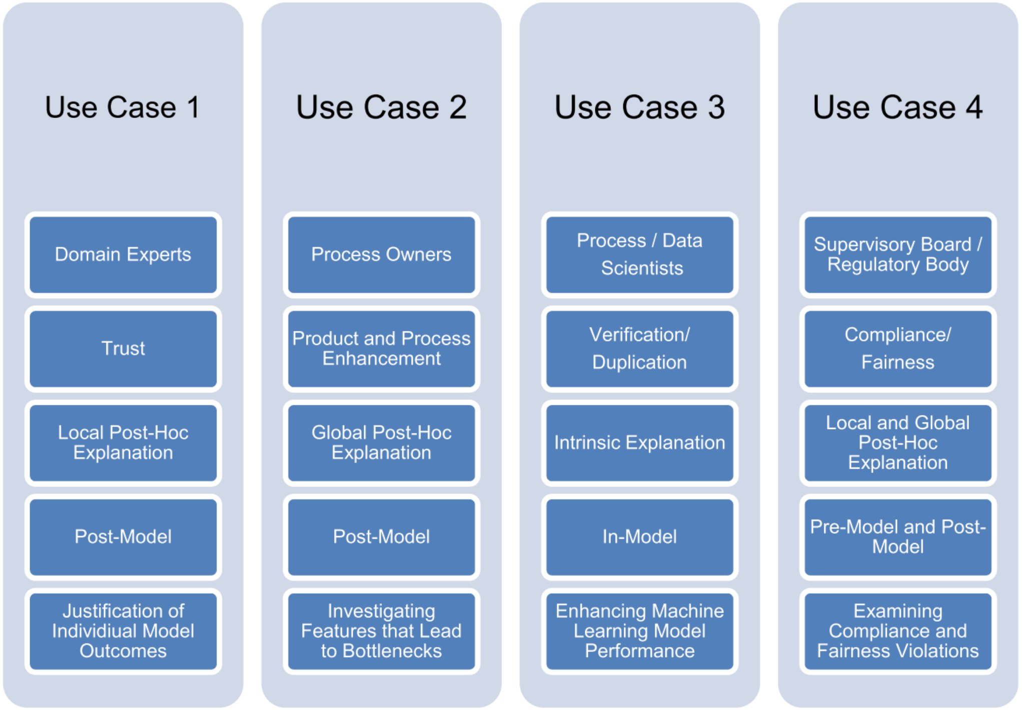
2.4 Related Work

From our best knowledge, we didn't find any relevant work concerning dashboard intended for business users. However, we outline gaps which exist in predictive process monitoring, which we might be able to correlate with during our development process:

Márquez et al. [21] on predictive process monitoring compiled a list of predictive process monitoring methods and techniques and provided recommendations for future research, which is current tool lags. Their research claimed that most of the tools and techniques being used for predictive monitoring fail to provide useful recommendations and explanations to the users and suggest that the concept of predictive monitoring is not yet widely applicable in the real world. Finding useful insights from real-world event logs is challenging since they are complex and require much effort to understand, so it is important to develop tools that help users query the models. They further also added that the recommendations must make sense to the user in the domain, which means that domain knowledge should be included in the analysis to identify all the potential recommendations, as most methods focus on improving the learning algorithm's accuracy rather than the quality of the predictions. These are the issues addressed in this thesis to ensure that the predictive process monitoring system is trusted and, as a result, is utilized by business users who are domain experts.

Nunes et al. [10] research on the explainable AI tools and methodologies identified multiple objectives like describing the system operation (transparency), assisting domain experts in making rational decisions (effectiveness), persuading users to invest in the system (persuasiveness), enhancing the simplicity of usage (satisfaction), enabling users to get knowledge from the system (education), assisting users in making quick judgments (efficiency), allowing users to notice system faults (debugging) and enabling user interaction with the system (scrutability). In the thesis, we tried to reach an objective intended for domain experts using our framework.

Mehdiyev et al. [9] proposed the use case for domain experts, who have limited expertise in machine learning and looks out for explanation from the individual observations. Figure 2.6 provided the overview of Domain expert use case (Use Case 1).



Use Case 1	Use Case 2	Use Case 3	Use Case 4
Domain Experts	Process Owners	Process / Data Scientists	Supervisory Board / Regulatory Body
Trust	Product and Process Enhancement	Verification/ Duplication	Compliance/ Fairness
Local Post-Hoc Explanation	Global Post-Hoc Explanation	Intrinsic Explanation	Local and Global Post-Hoc Explanation
Post-Model	Post-Model	In-Model	Pre-Model and Post-Model
Justification of Individiual Model Outcomes	Investigating Features that Lead to Bottlenecks	Enhancing Machine Learning Model Performance	Examining Compliance and Fairness Violations

Figure 2.6: Use cases for explainable process predictions, Taken from [9]

Since there is no prior framework developed considering the domain experts as their use case, our thesis will try to fill this gap and provide the first framework in this area.

Chapter 3

Dashboard Backend : Data Pre-processing and LSTM Model Improvement

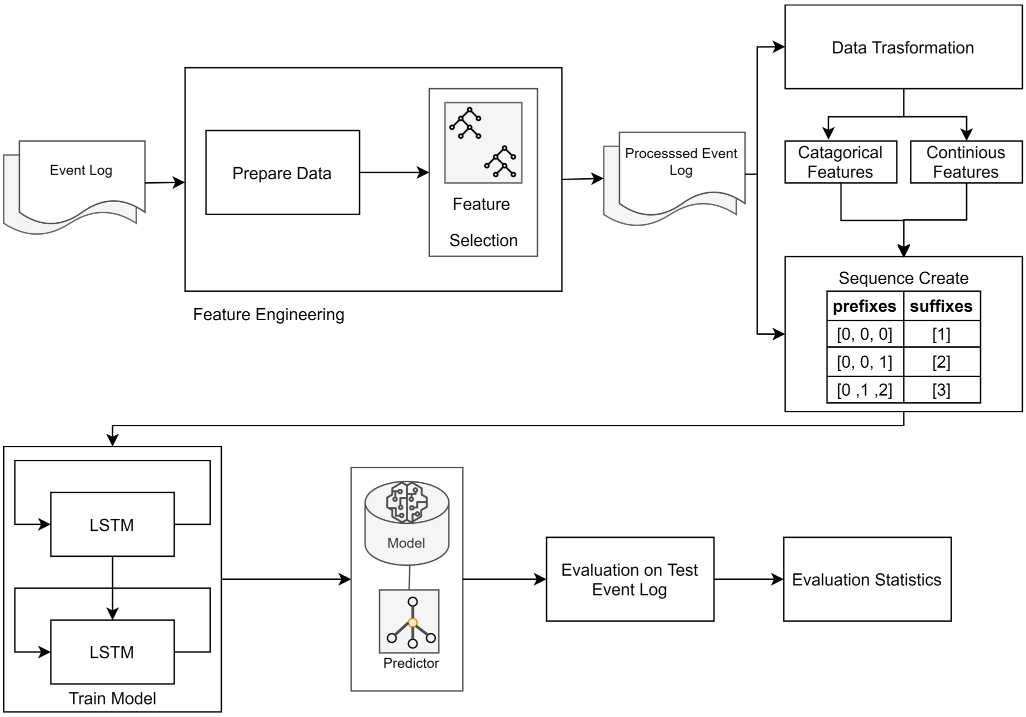
In this chapter, we will analyse the back-end part of the dashboard, especially on how the existing predictive model is improved and trained for the business process event log. The *Camargo et al.* model [22] LSTM architecture is taken as the baseline model and is compared with the improved LSTM model. The baseline model can predict each event's activity, role, and time, although, from our initial analysis, it lacks meaningfulness as it learns only the most frequent behaviour of events because of no contextual features incorporated during training. Our approach improves upon the LSTM model from Camargo's baseline model, allowing for a more accurate prediction of the Next event and an increase in the meaningfulness of the prediction regarding the event's log characteristics. The improved design is capable of reproducing the behaviour seen in the log more precisely. This chapter addresses the following research questions :

Research Question : How to improve the prediction of the running case for the selected LSTM base model for the given business process for more explainability?

This chapter is divided into the following sections — Section 3.1 deals with the overview of the training methodology, while Section 3.2 deals with the event log used and its associated statistics. Section 3.3 assesses the features used in training the model, and Section 3.4 explains how the features have been encoded. In Section 3.5 we explain the architecture design, followed that the experimental setup in Section 3.6, and the evaluation measures in Section 3.7. Lastly, in Section 4.2 we discuss the technique used to perform multi-prediction to address the second research question.

3.1 Overview of the Training of the Model

In this section, we briefly go through each step performed to train the model. As shown in Figure 3.1, we first select the event log and then impute the missing values in preparation of the data. Followed by this is feature engineering, which is performed using a tree-based classifier to select the important features. The pre-processed log is encoded and normalized for categorical and continuous features (numerical), after which the sequence is created, which will be learnt by the LSTM models. In the post-processing phase, the Next event has been predicted and evaluated for the event log (test log).



prefixes	suffixes
[0, 0, 0]	[1]
[0, 0, 1]	[2]
[0 ,1 ,2]	[3]

Figure 3.1: Train Architecture

3.2 Dataset Understanding: Features in the Source Data

This section describes the event log used in the experiment. Table 3.1 describes the brief statistics of the event log.

Sepsis Cases : This event log is publicly available at the 4TU Centre for Re-

Table 3.1: Statistics of the Event Log Used

event log	# cases	# activities	# events	case length		case duration (days)	
				avg.	max.	avg.	max.
Sepsis Cases	1050	16	15214	14.48	185	28.5	422

search Data[1]. This real-life event log records the trajectories of patients with life-threatening sepsis symptoms at a Dutch hospital. Each case records the events that were executed for one patient, from the time the patient checked in at the ER to the time he was discharged. Event attributes include, for example, the results of laboratory testing and the diagnosis of different symptoms in a binary form. Furthermore, each case ends when the patient is discharged. The features of the dataset contains Case ID, Activity, Complete Timestamp, User, CRP, Leucocytes, LacticAcid, Diagnose, Age, InfectionSuspected, DiagnosticBlood, DisfuncOrg, SIRSCritTachypnea, Hypotensie, SIRSCritHeartRate, Infusion, DiagnosticArtAstrup, DiagnosticIC, DiagnosticSputum, DiagnosticLiquor, DiagnosticOther, SIRSCriteria2OrMore, DiagnosticXthorax, SIRSCritTemperature, DiagnosticUrinaryCulture, SIRSCritLeucos, Oligurie, DiagnosticLacticAcid, Hypoxie, DiagnosticUrinarySediment and DiagnosticECG. The most reoccurring behaviour (20% arcs) of the event log, illustrated in Figure 3.2.

Figure 3.3 shows the BPMN model of the Sepsis Event log with 20% arcs. Starting from ER Registration until IV Antibiotic, there is a directly follow relation among the activities. At the same time, the concurrency occurs mostly among the LacticAcid, Leucocytes and CRP, which can occur in multiple orders after either Admission NC or Admission IC. Increasing the arcs in the process mining tool shows more concurrent behaviour, but it is impossible to show them in the report. As a result, the challenge for the LSTM-based prediction model will be to effectively manage concurrency in situations where activities may be executed in any sequence.

3.3 Feature Selection

3.3.1 Feature Selection of Baseline LSTM Model

In the base model, the authors used Activity, Role, and Timestamp as their main features. Embedding methods were used for preprocessing of activity and role, followed by sequence creation. In our improvement, intra-case and inter-case

[1] Sepsis Cases: https://data.4tu.nl/articles/dataset/Sepsis_Cases_-_Event_Log/12707639.

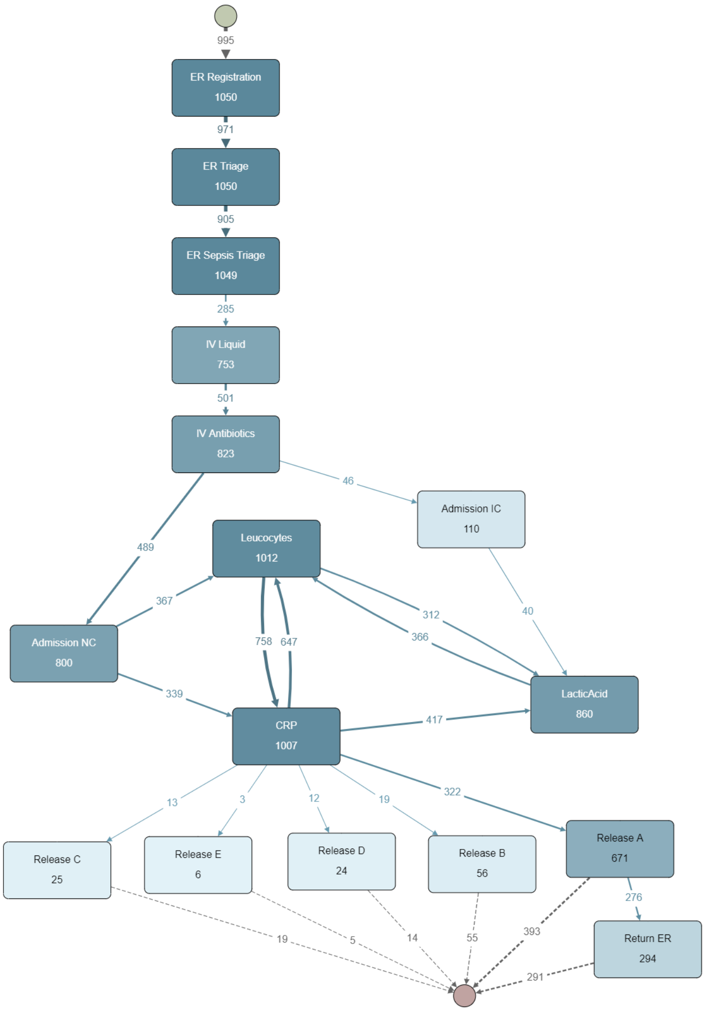


Figure 3.2: The most reoccurring behaviour of Sepsis Cases

features were used to extend the feature vector. The approach of selecting features has been discussed in the next subsection.

3.3.2 Feature Selection of Improved LSTM Model

The event log is first converted from *.xes* to *.csv*, and then the case and event attributes were identified for imputation of missing data accordingly. Case attributes have been kept consistent throughout a case. All the event attributes found were numerical in nature. So, for each case event attributes has been inserted with a zero until a value is found, which is populated for the rest of the event for the given case until there is a change in value. Incomplete cases were removed, i.e., in the Sepsis

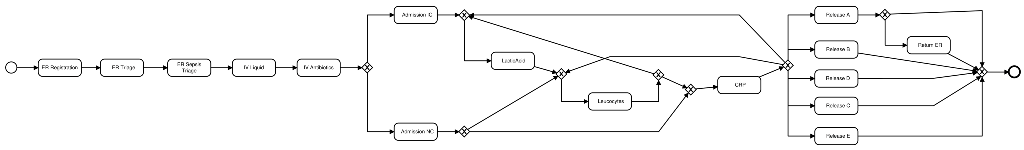


Figure 3.3: BPMN model of Sepsis Case (zoom in)

event log, if the trace does not have any of the following activities, it is removed: Release A, Release B, Release C, Release D, Release E.

Some recent studies have shown that adding characteristics derived from collections of instances (inter-case features) to predictive models improves the accuracy of the model's predictions [23, 24]. The number of "open" cases when an event is executed is reflected in an inter-case characteristic that has been exacted as a numeric dynamic attribute.

There are a predetermined number of potential values for each attribute. Certain attributes may have a wide range of values, with certain values occurring more often than others. Thus, to prevent the explosion of the dimensionality of the input dataset, only the values occurring at least for 10 instances are saved. Except for the activity (where all category values are used), this filtering is applied to every categorical attribute.

After processing inter-case features and removing infrequent attributes and incomplete traces, intra-case features have to be determined, which, along with the inter-case feature, will be used to encode it as an input for an improved LSTM model. In order to predict the next activity and the continuation of a running case, a tree-based classifier is incorporated to pick the most important intra-case attributes based on activity.

Random forest (RF) classifier is used to extract the importance of each intra-case feature with respect to the target variable (activity) due to its high accuracy in recent research [25]. A single tree-based classifier is straightforward; however, it is prone to output flipping under minor input changes as it learns only from one pathway, usually overfits, and lacks stability. The random forest method generates a large number of decor-related decision trees, which are then analysed. For each observation, all of these decision trees are processed, and the outcome with the highest probability of occurrence is utilised (i.e., bagging) as the final result of the model, which reduces the variance.

The features used by the base model are mandatory, so excluding timestamp, activity, role and also case-id, other features were used as an input feature to the model in order to rank the importance of the features with respect to the target output variable (Activity). The most influential attributes were selected based on the threshold, which is the mean value of the importance measure. In the Sepsis log, Leucocytes, CRP, LacticAcid, Diagnose and Age were equal and above the mean threshold of 0.037. The mean value is obtained by dividing the sum of the feature importance of each feature by the total number of features. The Diagnose and Age are the log's case attribute, while other selected features are Event Attributes. Figure 3.4 shows the input feature ranked in ascending order which is used for the classification. Additionally, meaningful features such as weekdays and hour of the day were extracted during the preprocessing of the timestamp, although these were not included in the selection of feature importance.

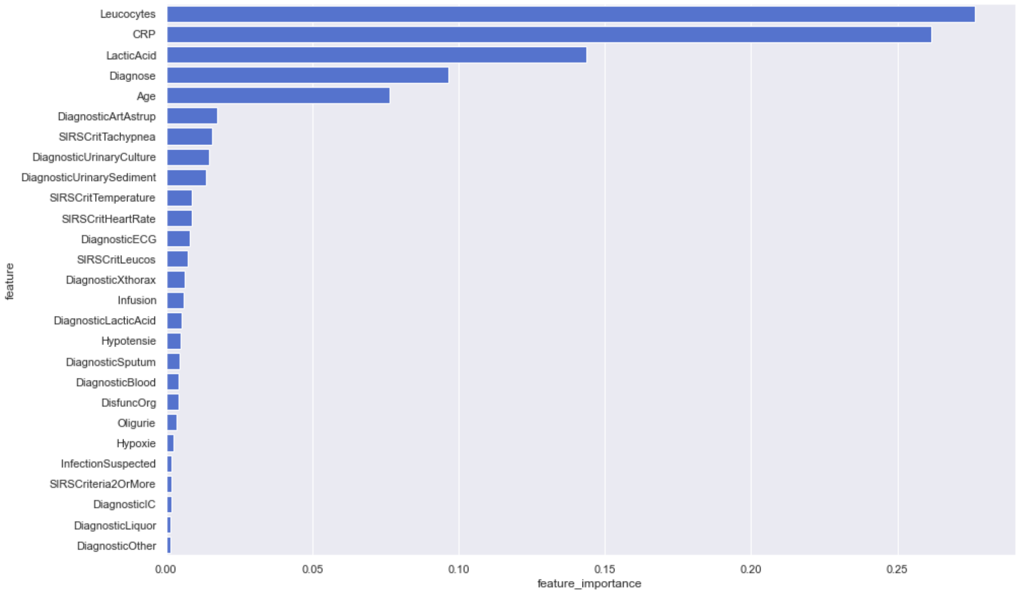


Figure 3.4: Intra-Case Feature Importance for Sepsis Event Log

The pre-processing steps taken are summarized in Figure 3.5.

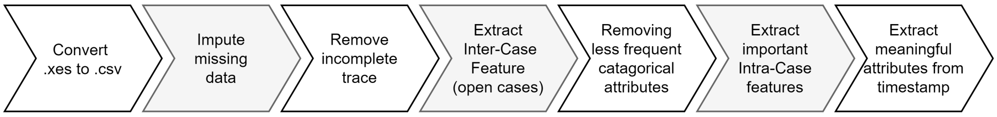

Convert .xes to .csv
Impute missing data
Remove incomplete trace
Extract Inter-Case Feature (open cases)
Removing less frequent catagorical attributes
Extract important Intra-Case features
Extract meaningful attributes from timestamp

Figure 3.5: Pre-processing steps

Table 3.2 lists all the features which were incorporated along with the mandatory attributes and were transformed into feature vectors to perform prediction from the LSTM model.

Table 3.2: Sepsis Feature grouping and its type

Feature Group	Name	Category	Attribute Type
Mandatory	Activity	Categorical	Event Attribute
	Role	Categorical	Event Attribute
	Timestamp	Numerical	Event Attribute
Inter-Case	Open Cases	Numerical	Event Attribute
Intra-Case	Leucocytes	Numerical	Event Attribute
	CRP	Numerical	Event Attribute
	LacticAcid	Numerical	Event Attribute
	Diagnose	Categorical	Case Attribute
	Age	Numerical	Case Attribute
Timestamp	Weekday	Categorical	Event Attribute
	Daytime	Numerical	Event Attribute

3.4 Feature Encoding

3.4.1 Feature Encoding of Baseline LSTM Model

The feature encoding of *Camargo et al.* [22] model is divided into two main parts — handling of feature type and sequence creation. The handling of feature type is subdivided into transforming categorical and numerical attributes in the predictive model interpretable format.

In the handling of the mandatory categorical attributes (activity and role), the number of roles was first reduced by grouping them together using the algorithm described by Song and Van der Aalst [26]. The algorithm tries to discover roles by creating a correlation matrix based on similarities between activity execution patterns and the users who executed it. This reduced the number of categories for the role while retaining enough information to enable the LSTM network to distinguish between events. The authors then used the embedding method (2.3.1.2) to encode the activity and role. The generated embeddings were used across the experiments with different models.

Numerical attributes have to be scaled in the range of [0,1] to be interpreted by predictive models. The timestamp is converted to relative time between events and is calculated as the time elapsed between completion of the event with the previous event. Next, it is normalized either by max-normalization or log-normalization to balance the data's high and low variability.

Since the LSTM based predictive models take sequence-based inputs, n-grams were used of fixed size to create an input sequence to train the predictive model. It allows the control of events in the time-sequenced input and helps map them into a pattern of n sized sub-sequences, which helps the model understand the order of execution regardless of trace length. N-gram is formed at each time-step of the execution for all the attributes independently after it has been normalized according to its respective categories such that it is understandable by the predictive model, i.e., for activities, roles, and time it is being sequenced independently. The table 3.3 tabulates the five n-grams of the first five events of case id DHA in the sepsis event log. The numerical representation of activity, role, and time corresponds to the indexes and scaled values that were created throughout the data transformation phase.

Table 3.3: N-grams for case number DHA from the sepsis event log

Time Step	Activities	Roles	Timestamp
0	[0 0 0 0 0]	[0 0 0 0 0]	[0. 0. 0. 0. 0.]
1	[0 0 0 0 4]	[0 0 0 0 2]	[0. 0. 0. 0. 0.]
2	[0 0 0 4 6]	[0 0 0 2 5]	[0. 0. 0. 0. 2.08314159e-05]
3	[0 0 4 6 5]	[0 0 2 5 2]	[0. 0. 0. 2.08314159e-05 4.16073554e-07]
4	[0 4 6 5 10]	[0 2 5 2 4]	[0. 0. 2.08314159e-05 4.16073554e-07 0.]
5	[4 6 5 10 3]	[2 5 2 4 4]	[0. 2.08314159e-05 4.16073554e-07 0. 0.00010715]

3.4.2 Feature Encoding of Improved LSTM Model

The inter-case and intra-case features, which have been identified in section 3.3.2, are converted into feature vectors along with the mandatory features. The processing of mandatory features remained the same as the baseline LSTM model, as discussed in section 3.4.1. In improved model encoding, categorical features are processed using one-hot encoding (2.3.1.1). One-hot encoding assigns the target feature the value 1, and the other features the value 0. To put it differently, if N is the attribute's cardinality (the number of unique values or levels), then one-hot encoding maps them in 1-to-N [27].

In the sepsis event log, Diagnose and Weekday are categorical features, which were converted to feature vectors using the one-hot encoding. The rest of the features are numerical, and the normalization techniques used in the base model has been used to transform them.

After the respective encoding of each feature, they are concatenated. Following this, the sequence generation step is executed to extract n-grams for controlled handling of long traces. Each step of the process execution generates one n-gram, and this is done independently for each attribute. The improved model uses four inputs: prefixes of activity, prefixes of role, prefixes of relative time, and extracted concatenated contextual attributes (inter-case, intra-case, and timestamp extract) with the expectation that it will learn a meaningful pattern for better prediction.

3.5 Model Structure

Since LSTM networks are a well-known and established technique for handling sequences, which are the essence of a business process event log, they serve as the foundation of prediction models used. The architectures suggested by *Camargo et al.*, are based on the Figure 3.6.

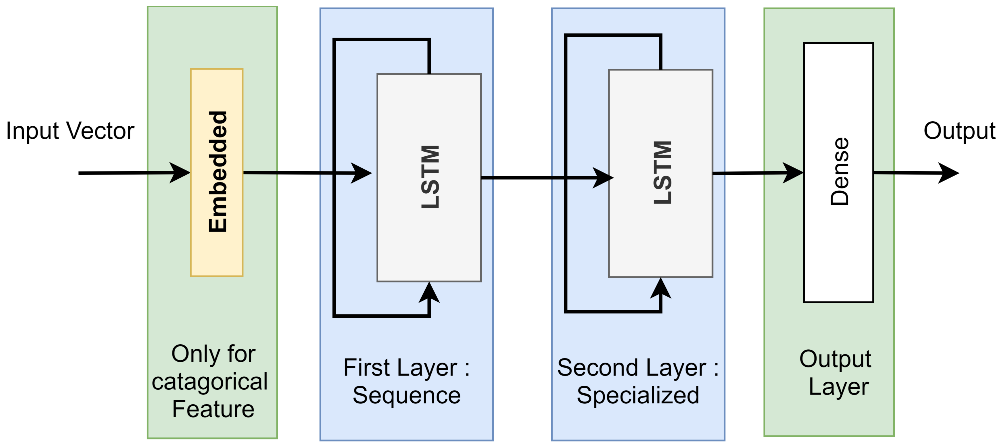


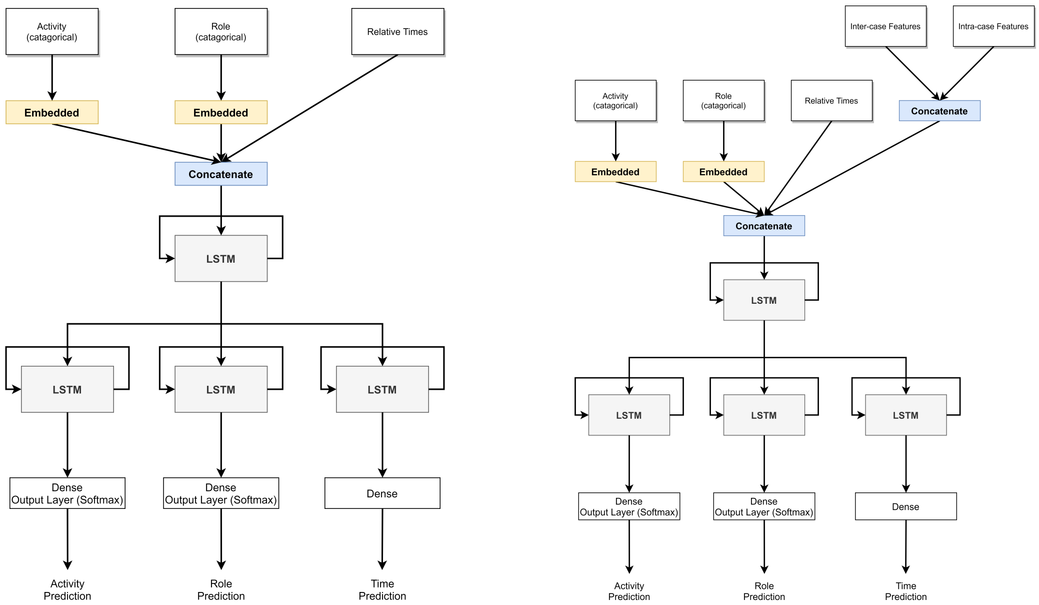
Figure 3.6: Generalized Architecture of the LSTM

They consist of an input layer that takes up each attribute individually; for the mandatory category, attributes (activity and role) have an embedding layer. It is followed by two stacked LSTM layers and a dense output layer. The first LSTM layer outputs a sequence rather than a single value to the second LSTM layer. The second layer is specialized, each for individual activity, role, and relative time predictions.

The architecture discussed in *Camargo et al.* differs depending on if the first layer is shared. The intuition behind shared layers is that it aids in identifying better execution patterns. Thus, we have only used the full shared architecture, which concatenates all inputs and shares the first LSTM layer entirely as shown in Figure 3.7a as the base model. As for improved model along with the base model input vectors, concatenated contextual features extracted (3.4.2) were incorporated as shown in Figure 3.7b.

3.6 Experimental Setup

The evaluation's objective is to assess the accuracy of the baseline model to that of the improved model discovered from event logs. The evaluation metrics used



(a) Base LSTM Architecture

(b) Improved LSTM Architecture

Figure 3.7: LSTM Network Architecture

are standard in prediction model research, which are also used in the *Camargo et al.* [22]. The quantitative metrics helps to determine the accuracy and to determine the meaningfulness evaluation has been extended with replay analysis using process mining tool (Apromore). The experimental pipeline followed is displayed in Figure 3.8.

With the hold-out technique using the temporal split criteria, the event log was divided into two partitions, with 70% of it utilized for training and 30% used for testing. Temporal splits are often used in predictive process monitoring since it helps to avoid information leakage, preserving the order between the cases [6, 28]. The first 80% of the training data is used for model training and the remaining 20% for validation.

The parameters used during the experiment are tabulated in Table 3.4. Also, early stopping is used to avoid overtraining. The best model for both base and improved models was achieved by tuning these parameters.

However, while training the model, it was found that it was giving rise to overfitting upon including all the features to form the improved model. So, along with tuning the model parameters, different combinations of intra and inter-case attributes were also trained. The learning curve of the trained model is shown in Figure 3.9. In the sub-Figure 3.9b it is evident that the distance between validation and test curve is quite evident.

The features from Table 3.2, which were not contributing much during the

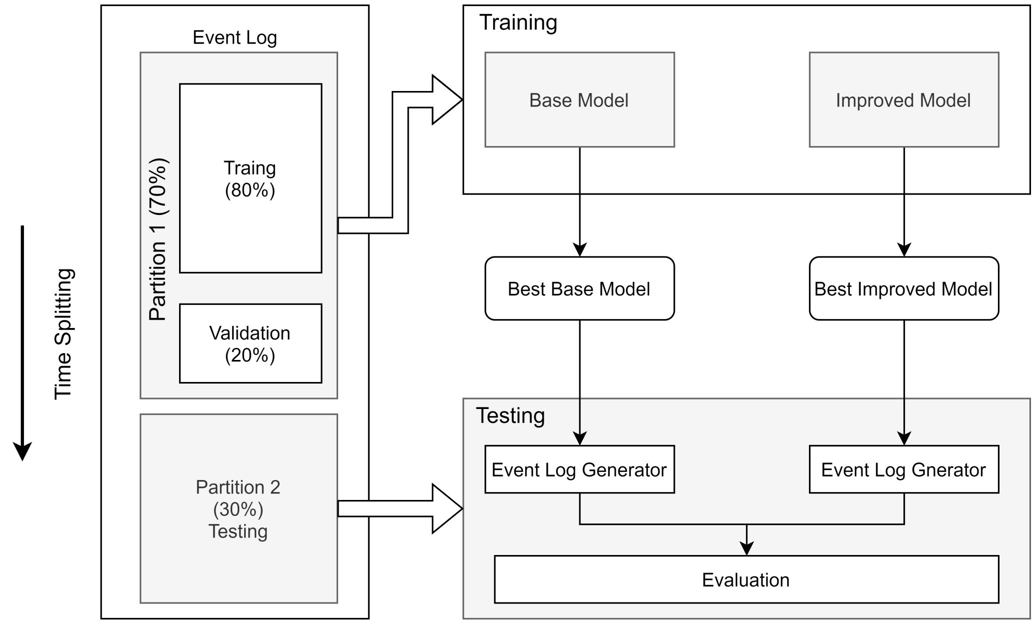
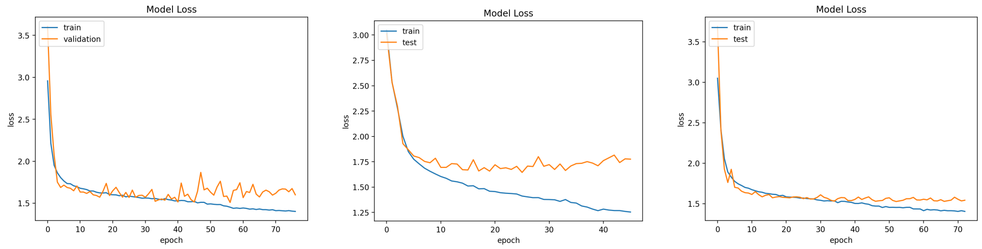


Figure 3.8: Experiment Pipeline

Table 3.4: Parameters used for finding the optimal Model

Parameter	Values
Batch Size	[32, 64]
N-gram size	[5, 10, 15]
Method for Numerical Input Scaling	[Max, Lonormal]
# units in hidden layer	[50, 100]
Optimization Function	[Nadam, Adam]
Activation Function for Hidden Layer	Tanh



(a) Baseline Model (b) Full Improved Model (c) Opt. Improved Model

Figure 3.9: Train and Validation Learning Curves

experiment, were removed to achieve better accuracy. The optimized improvement model includes these inter and intra-cases: Open Cases, CRP, LacticAcid, and Leucocytes, while the full improvement model includes all the attributes mentioned in Table 3.2. After performing multiple experiments, we found the optimal model parameters which can be used to train each model, and it has been described in Table 3.5. The evaluation section will evaluate the three models: baseline model, improved model (full), and improved model (optimized).

Table 3.5: Optimal Model Parameters

Batch Size	N-gram size	Method for Numerical Input Scaling	# units in hidden layer	Optimization Function	Activation Function for Hidden Layer
32	10	Max	50	Nadam	Tanh

The models are evaluated throughout the testing phase by continuously providing the model with sequences of prefixes of varying lengths from the test log. It predicts the subsequent events repeatedly until the end of each individual case, after which we capture the whole next event generated for each row to evaluate. In the evaluation, we only incorporate the maxima, i.e., arg-max. Arg-max is a method of choosing the prediction category with the maximum predicted probability, and it performs well for tasks like predicting the most probable next event given an incomplete case.

3.7 Model Evaluation

3.7.1 Evaluation Measures

In this section, we incorporated two broad evaluation measure, qualitative and quantitative. Qualitative assessment is carried out by replaying the generated predictions using a process mining tool and comparing them to the test event log behaviour. Quantitative analysis is performed not only by looking at accuracy measurements, but also by employing two metrics to assess the similarity between the original event log and generated event log.

Damerau-Levenstein (DL)[2] distance is applied to measure the similarity metrics for sequences of activity and roles [29]. It is a measure for evaluating sequences regarding the number of edits required to make one string character identical to another, i.e., how dissimilar two strings are. Insertion, deletion, substitution, and transposition are all supported edit procedures. In the case of two parallel activities,

[2] Damerau-Levenshtein Distance: `https://jellyfish.readthedocs.io/en/latest/comparison.html#damerau-levenshtein-distance`.

transpositions are permitted without penalty. This means that they may appear in any order, i.e., given two activities, we can see both [A, B] and [B, A] in the log. The resultant Damerau-Levenshtein distance is then normalized by dividing the total number of edit operations by the length of the longest sequence. Hence, its inverse is used (subtract the normalized DL distance from 1) to compare a produced sequence of activities or roles to an observed sequence in the event log. A higher value indicates more similarity between the sequences.

Measurement of the inaccuracy in predicting timestamps is done using the Mean Absolute Error metric (MAE). It is measured first as the difference between the cycle times of observed and the predicted paired traces t1 and t2, and then computing the average over a collection of paired traces. A smaller value indicates less difference between the sequences.

3.7.2 Quantitative Evaluation

The experiments were performed for the sepsis dataset on the Camargo LSTM architecture, i.e., baseline model, since it was not used as one of the datasets in the original research, followed by on the Improved Model (Full) and Improved Model (Optimized).

Table 3.6 tabulates the performances of the three models. Overall, both the improved models perform better than the base model except in the case of Role distance similarity in the quantitative evaluation measurement. MAE performed significantly better for the full Improved Model, and slight improvements were seen in the optimized model. Moreover, the similarity measurement for the Activity is better for optimized, improved model.

Table 3.6: Experimental results for the Sepsis Cases

Measure	Base Model	Improved Model (Full)*	Improved Model (Optimized)**
Activity Accuracy	0.5392	0.5211	**0.5763**
Role Accuracy	0.7937	**0.8046**	**0.7990**
DL Activity	0.3601	0.2636	**0.4706**
DL Role	**0.8564**	0.8247	0.8558
MAE (Days)	2.8624	**1.6792**	**2.6583**

* Model encoded with all the features from the Table 3.2

** Model encoded with the Open Cases, Leucocytes, and Age

3.7.3 Qualitative Evaluation

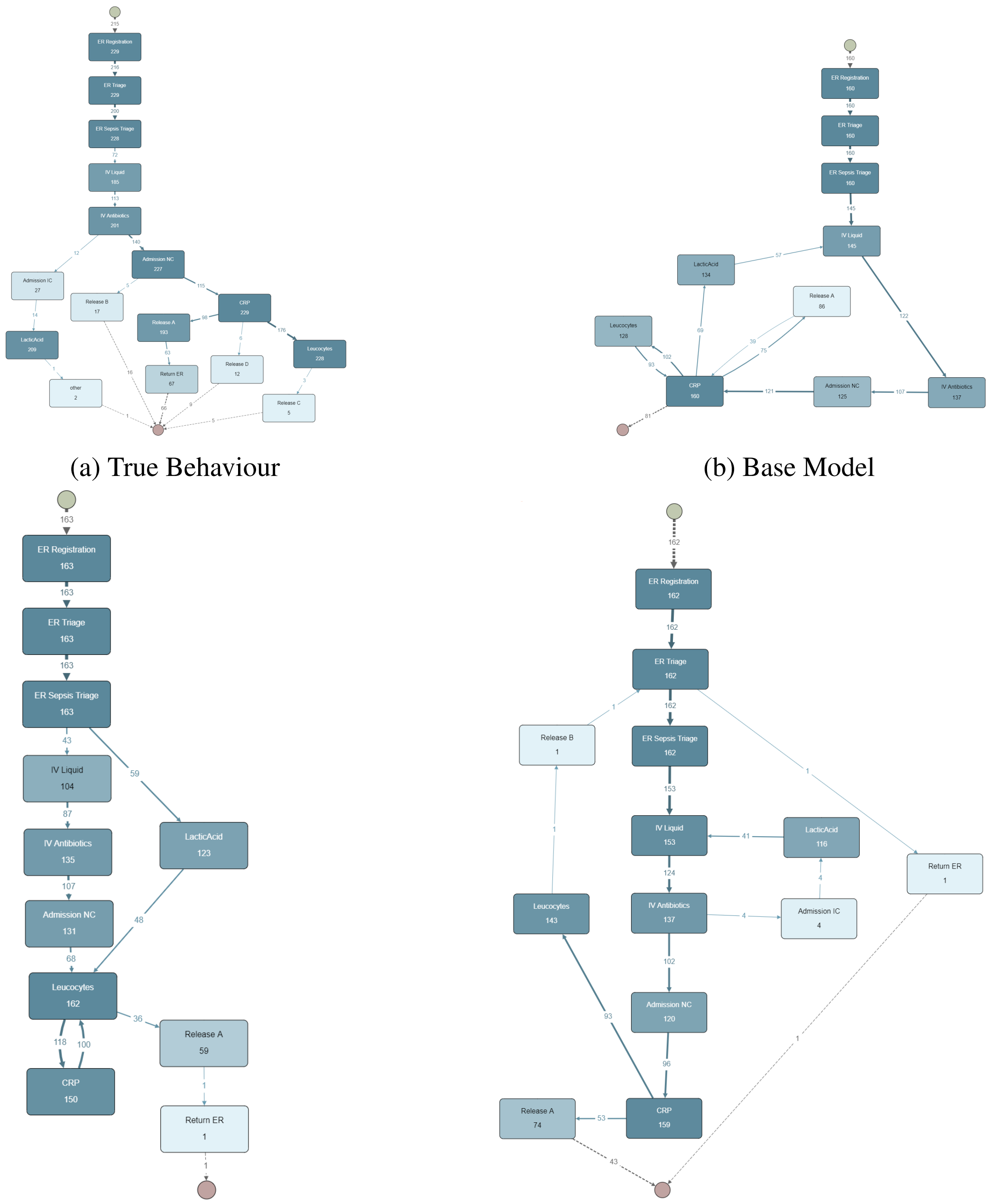
The qualitative evaluation involves a process mining tool (Apromore) to visualize the event log found using the three models and estimate how much it varies from the ground truth event log. The LSTM model used generates activity, role and expected duration (time) individually. The test event log was replayed over the model, and the predicted outcome for each event was captured. The prediction of each event was converted back to readable form, which resulted in a generated event log for each model.

Figure 3.10 shows the directly follow graph (DFG) of sepsis for ground truth vs that generated from different models with top 10% behaviour (arcs). The first striking observation among ground truth vs all DFG is that it did not generate all the activities. The correct order of activity Return ER is learned by the improved model (full) (Fig 3.10c), whereas the improved model (optimized) learns it but not in the correct order. The base model, on the other hand, does not have this activity at all. None of the models learned the release type Release C and Release D. Although all the models learned the Release A, but the Release B was only learnt by the improved model (optimized) (Fig 3.10d).

The evaluation reveals that the LSTM encoded with optimal attributes does perform very well for the initial activities, i.e., ER Registration, ER Triage, ER Sepsis Triage, IV Liquid and IV Antibiotics, as shown in Figure 3.10d. It is also able to predict traces for intensive care activity, i.e., Admission IC, which is followed by LacticAcid, which is in accordance with the true behaviour (Fig 3.10a). Other models were not able to learn this entirely.

The activities that follow normal care, i.e., Admission NC, CRP and Leucocytes, are simulated well by the Improvement model (Full) (Fig 3.10c), as it leads to the proper release of the patient with Release A and Return ER. Improvement model (Optimized) (Fig 3.10d) is not quite correct in the perspective considering activity that follow the Release A. The base model (Fig 3.10b) captures much rework between activities and has arcs connecting to the release of the patient. Overall, all the models performed impressively in predicting initial activities accurately but lacked in predicting the later events.

The quantification of qualitative evaluation can be done using conformance checking and determining the fitness of each event log with respect to the ground truth (test event log) process model. Process Mining tool (Apromore) offers to save the BPMN model of the logs and calculate the fitness. Fitness measures the extent to which the discovered model correctly mimics the log entries [30]. The ground truth BPMN model was extracted with 10% of the arcs, and the respective fitness is tabulated in Table 3.7. The Improved model (Optimized), improved model (Full) and baseline model outperform the ground truth behaviour in terms of fitness.



(a) True Behaviour

(b) Base Model

(c) Full Improved Model

(d) Opt. Improved Model

Figure 3.10: Directly Follow Graph of Sepsis on Next Activity Prediction

Table 3.7: Model Fitness Measure

	True Behaviour	Base Model Behaviour	Improved Model (Full)	Improved Model (Optimized)
Fitness	0.266	0.346	0.275	**0.359**

3.8 Summary

This chapter discussed the system's backend and the existing predictive process monitoring LSTM architecture and how it was improved using the Sepsis Event log. Quantitative and Qualitative evaluation was used to discuss how it performs with respect to the ground truth. In the rest of the thesis, the work is presented with the Improved Model (Full) as it performed competently in qualitative measure among all the models, and the evaluation of the multi-prediction will be performed using this model in Chapter 6. In the next chapter, we will go through the overview of the system, its conceptual design and the feature it offers.

Chapter 4

Dashboard Architecture Overview and Features

The project's first objective is to formulate a dashboard framework that can emulate the predictions generated by a deep learning-based (in this case RNN-LSTM) prediction model for the ongoing case of a business process, which can assist business users in making informed decisions. In this chapter, we will go through the requirements, developed system design, technology, and the different features it offers and addresses the overview of following research question :

Research Question : How should the dashboard be designed to incorporate the LSTM models to simulate the predictions generated for the users?

We will also discuss the multi prediction method and answer the following research question :

Research Question : How will the model provide alternate recommended actions for an event that could lead to a successful outcome?

The chapter is divided into the following sections, in Section 4.1 we formulate the requirements which are needed for the dashboard, after this section we start formulating the sections how these requirements met. The Section 4.2 discusses how the existing LSTM model would be able to make the multi predictions, following that in the Section 4.3 we discuss the system architecture and then the Section 4.4 discusses on the User Interface conceptual design and the technical framework which has been used. Following that, it in Section 4.5 it briefly introduces the different feature the dashboard offers and lastly in Section 4.6 discusses upon the different control options available for those features.

4.1 Functional Requirements

The objective of the functional requirement is to model the application, keeping a specific set of users in consideration. Mehdiyev et al. identified those users and segregated them into a different use cases. The requirements in our work are more aligned towards the users of who are domain experts (use case 1) with little experience with predictive models who simply wants to utilize the explanations supporting the model recommendation for specific observations [9]. The user should be able to visualize enough information, making an informed choice for an ongoing case. This section lists out the functional requirements of the dashboard, considering the research objective and the users it is intended to :

1. The dashboard should support multi-predictive recommendations to provide user the possible options which can be taken over the state of the process. Thus, the views and their respective modes will support the multi-predictive recommendations of activities and roles and will show confidence for each recommendation. This may lead to change the output layer of the LSTM model.

2. Views for different prediction processing, i.e., batch processing and single event, should be separate. Single event processing would be used for predictive recommendation of an ongoing case, whereas batch processing would be used for evaluation of the techniques for evaluating different predictive recommendations.

3. The batch view would be used to evaluate the predictive recommendations employed over the test event log. Unlike model evaluation, here it should evaluate among the multi-predictive recommendations of different predictive techniques.

   (a) It should support generating the predictions of the entire event log, with and without pre-selected prefixes.

   (b) Evaluation techniques such as Control-Flow Log Similarity (CFLS) using Damerau-Levenshtein distance, Mean Absolute Error (MAE) of cycle times and Event Log Similarity (ELS) should be integrated to have a better understanding of how much trust user can have over the model recommendations [22, 31].

4. The single event view should simulate the processing of an ongoing case.

   (a) It should support a mode which generates the predictive recommendations of the ongoing case one by one and show the history of actual prefixes taken into action so far along with the properties of the event by reflecting upon additional attributes present in the log.

(b) Another mode should provide assistance to simulate already executed cases where the length of the prefix trace can be maneuvered forward and backward, and it should support how the trace will turn out in the end based on the provided trace prefix and conduct a conformance check over the prefix to visualize how reliable the predictions are.

(c) It should support a mode which supports the simulation over the trace based on the selected recommended actions over the different action choices.

(d) It should support the labelling on the generated predictive actions performed after completing a certain number of events into a regular or deviant trace in all the modes above. The labelling is chosen by the user while replaying the events (at runtime) which shouldn't involve modification on LSTM architecture.

The requirements listed not only cover the users it is intended for, but also covers some part of other use cases in as well [9] owing to a significant direct and contextual relationship. The explanation provided in the following chapters and sections will only consider domain experts.

4.2 Multi Prediction from the Model

The LSTM model discussed in Chapter 3 need to generate multi-prediction and the confidence from the model as per Requirement 1. It is achieved by using the Softmax function in the dense layer (shown in Figure 3.7) for categorical attributes and extracting the prediction by sorting it based on the confidence (probability) the model has. In multi-class classification (in case of predicting activity and role), the Softmax function gives probabilities for each class. The output probability ranges between 0 and 1. Mathematically :

$$f(x_i) = exp(x_i)/\sum_{j=1}^{n} exp(x_j)$$

and it translates to calculating each class exponent and dividing it by the total of the exponent class values [32].

Numerical attributes (Timestamp) cannot have the multi-prediction because it is not a classification problem. Although, if the event prediction is fed individually to the model, then it will have different time predictions in each prediction.

As mentioned earlier, sorting of predictions is based on the probability measure, and it is either done by using maxima (Arg-max) or by random choice. While using the model in a generative way, it could be biased in the case of solely using maxima because it tends to generate a similar kind of sequence as it was sorted based on the most probable ones. This could be avoided using by using random selection of values from the categorical attributes being predicted, generating different traces, and avoiding being stuck in higher probabilities, although the experiment may have to be performed multiple times to get concrete results. Additionally, this approach enables to determine what the neural network has really learnt from the event log while training [22]. Thus, both approaches have been used while evaluating generative predictions, i.e., when there is no feedback to the model from the test log.

The number of predictive recommendations generated has to be capped, but also be dynamic based on the event log used. The activity and roles are predicted individually by the model but are capped in our design using the Algorithm 1.

Algorithm 1 Capping of number of Predictions

```
Require: $ac \leftarrow$ Number of unique activity
Require: $rl \leftarrow$ Number of unique roles
Ensure: $pr = 0$
  if $ac > rl$ then then
      $pr \leftarrow rl$
  else if $rl \geq ac$ then then
      $pr \leftarrow ac$
  end if
  Number of Predictions $\leftarrow pr$
```

4.3 System Overview

Figure 4.1 describes the overall system flow developed. There are two major components in the designed system: training and evaluating the model for the chosen event log and developing the dashboard around the model, which domain experts can use. The user interacts with the user interface in the browser, which sends *http* requests to the webserver, which then forwards the request to the application, to either train the processed event log, or to process the predictions.

The training of the LSTM models has been discussed in Chapter 3, where it improves the accuracy and meaningfulness of the *Camargo et al.* [6] full shared LSTM architecture. The model is improved by introducing inter-case features, which are obtained from feature engineering. Although process until feature engineering (Fig 4.1) is performed offline, but a view is created to tweak between the training parameters (Table 3.4) to train the model via the dashboard, although it is beyond the scope of this thesis and therefore, has not been discussed. The LSTM model generated from the training phase is saved and used along with predictor.

The two types of prediction processing can be accessed by selecting their respective view options; under which, it can access different modes. Single event processing offers Execution, Evaluation and What-If mode, whereas the Batch processing offers Base and Pre-select prefix mode. This information is communicated with the predictor, which fetches the predictions using the prefixes from the event stream (test event log) and the trained model. It returns the *http* response to the webserver to show the desired view and the data requested.

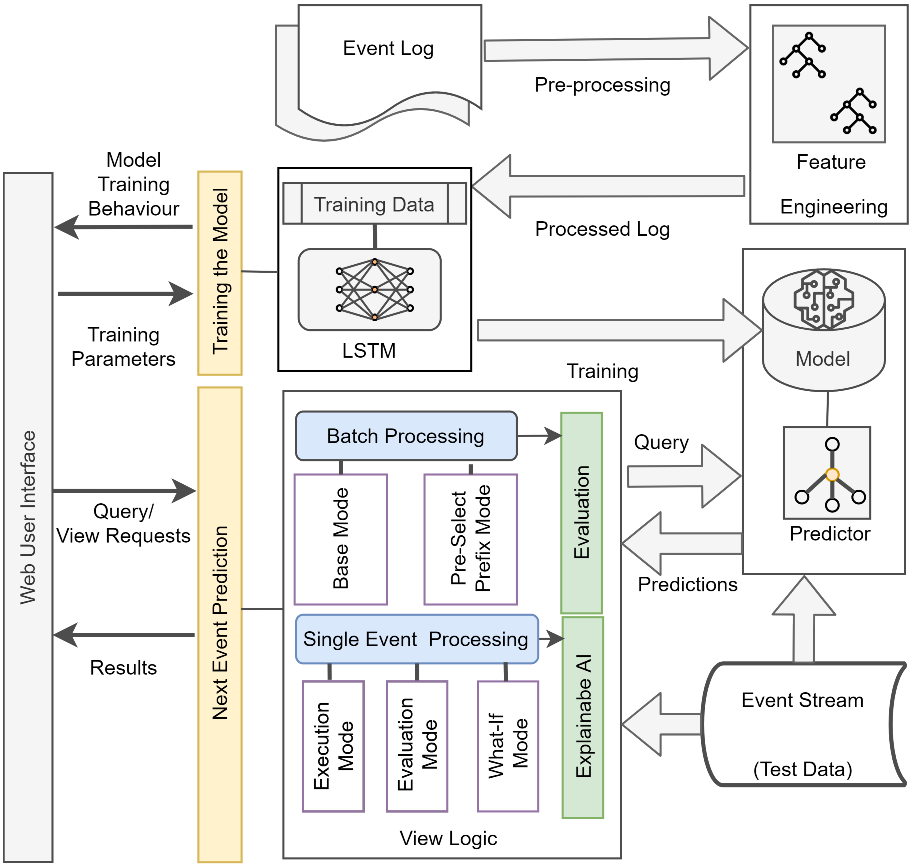
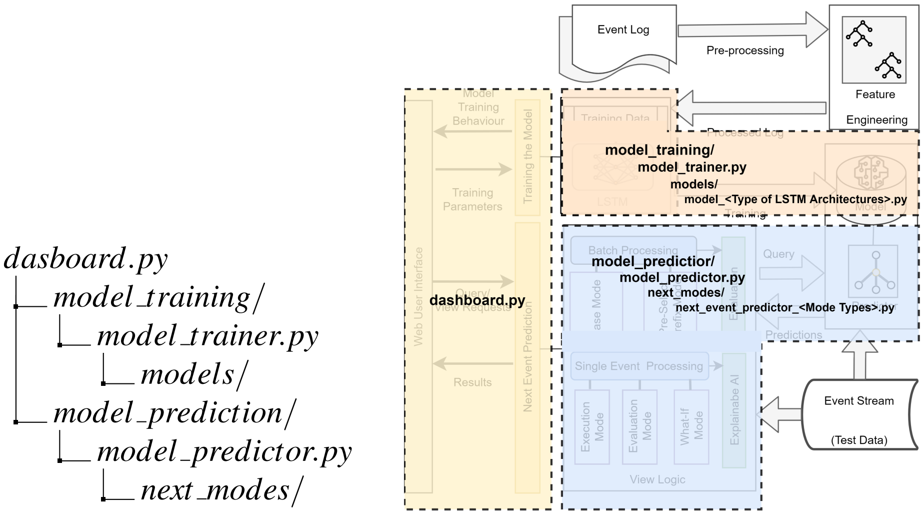


Figure 4.1: System Flow

4.3.1 Design Pattern

The design of the system roughly follows the Model-View-Controller (MVC) [33] framework, which is mostly used in interactive applications. In the MVC framework, the Model component holds the essential functions and the data, the View component displays information to the user, and the Controller component evaluates the user input. The user interface comprises the View and Controller [34].

Figure 4.2 shows the file structure of the framework developed (Fig 4.2a) and which component in the overview architecture it represents (Fig 4.2b). The `dashboard.py` contains both the View and the Controller for training as well as for the prediction, and it is where the application is put together. `model_training/` and the `model_prediction/` folders have the essential 'model' elements for training and prediction, respectively. In the case of prediction, there is an overlap of view and model, which the `model_predictor.py` encapsulates. The repository for the project is hosted on GitHub[1] and a trimmed-down version of the dashboard is hosted on the web[2] with public access.



(a) File structure

(b) Framework File Overview

Figure 4.2: Framework File Structure

[1] GitHub Repository: `github.com/rhnfzl/business-process-dashboard-for-lstm`.

[2] Streamlit Share: `share.streamlit.io/rhnfzl/ppm-dashboard/dashboard.py`.

4.4 User Interface

As per Requirement 2, the developed web application interface Next Event Prediction has two main views, — batch processing and single event processing. A common conceptual design for different modes for both views will be discussed in subsection 4.4.1, followed by the working of the programming framework used to build the dashboard in subsection 4.4.2.

4.4.1 Conceptual Design of User Interface

Following the Requirement 3 and 4, several modifications were made in both the views during the continuous development process, including the design of each view, the presented process information on views, the control menu, and the representation of the overall application. However, the programming framework used throughout the project to create the required user interface remained consistent.

Figure 4.4 represents the conceptual view of the developed dashboard for single event processing, and Figure 4.3 shows the conceptual view for batch processing. The common element among both views is the hide-able control section (App Control Menu) on the left-hand side, each control option has been discussed in Section 4.6.

The Batch Processing is designed to evaluate different predictive recommendations over the test event log, and the main screen offers the following options:

i At the top, a dynamic table where the processed test event log can be sorted and filtered.

ii Following that, a Quantitative metric table which displays the Damerau-Levenshtein distance measurement of Activity and Role, Mean Absolute Error (MAE), Control-Flow Log Similarity (CFLS) and Event Log Similarity (ELS) of different predictive recommendations.

iii Finally, comparison of each quantitative metrics among different predictive recommendation is visualized.

Single Event is designed to process each case one by one it has, and the main screen offers the following options:

i At the top across the different modes, it let the user select the case-id which is from the event stream (test log).

ii Following that, it offers the slider to maneuver the prefix of an executed case in the Evaluation mode.

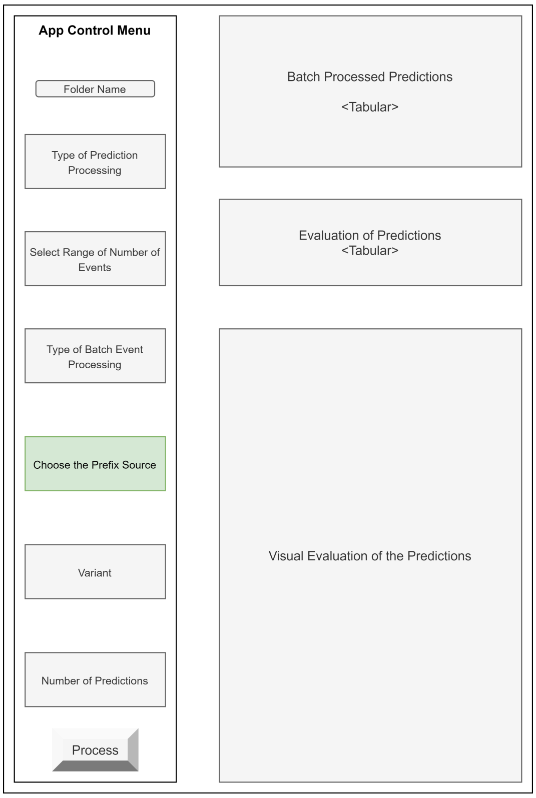

App Control Menu
Folder Name
Type of Prediction Processing
Select Range of Number of Events
Type of Batch Event Processing
Choose the Prefix Source
Variant
Number of Predictions
Process
Batch Processed Predictions
<Tabular>
Evaluation of Predictions
<Tabular>
Visual Evaluation of the Predictions

Figure 4.3: Batch Processing : View and Controls which are*Common* are in grey, and specific to *Pre-Select Prefix* mode in green

iii On the side of the slider, the control box is meant for What-If mode, where it let user choose the action from the recommendations.

iv After that, predicted recommendations are displayed where it groups the Activity, Role, and Time.

v Each of the elements of predicted events (activity and role) will be displayed with the confidence the model has on the recommendation; Figure 4.5 shows how it will be display if the model predicts the activity ER Registration.

vi At the end, in the Evaluation mode, a tabular representation of the conformance check visualization is displayed.

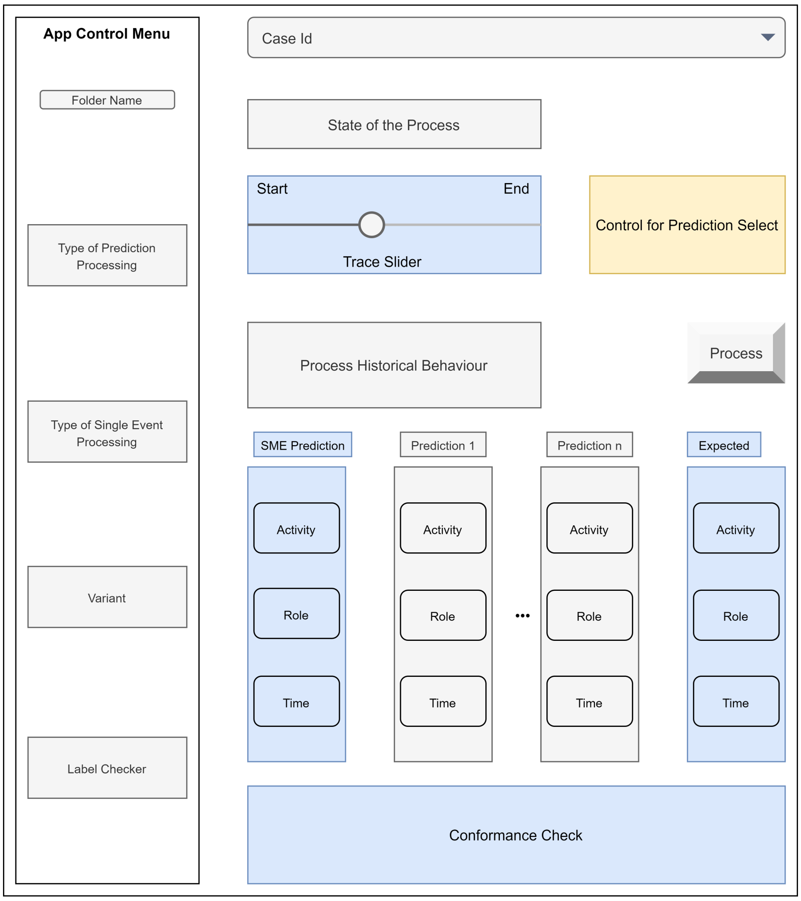


Figure 4.4: Single Event Processing : View and Controls which are *Common are in grey*, and specific to *Evaluation* and *What-If* mode are in blue and yellow respectively

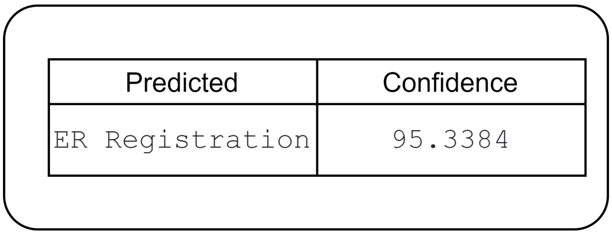
4.4.2 Overview of Streamlit Framework

Streamlit[3] is an open-source micro-web Python package. It assists in creating highly interactive web applications with existing data science and machine learning projects.

It features a distinct data flow, i.e., it executes the entire Python script top to bottom whenever anything on the screen needs to be changed, which can be due to a change in the source code of the application or the interaction with the control by the user. In short, there are no *callbacks*.

The built-in API's help to readily display data in a textual, tabular, or graphical form. The widgets, i.e., sliders, buttons, drop-down etc., helps to control the

[3] Streamlit : `docs.streamlit.io`.



Predicted	Confidence
ER Registration	95.3384

Figure 4.5: Conceptual design for displaying the predicted elements for an event

interactions, whereas the layout panel helps to organize these controls if required. It also provides a cache primitive that acts as a steadfast and immutable data storage, permitting the reuse of data without loading it at every execution.

So, to sum it up as shown in Figure4.6 :

- For each user interaction, the whole script is re-executed from top to bottom.
- Based on widget states, it assigns an up-to-date value to each variable and renders its output in real-time in the browser.
- It uses caching to prevent having to recompute time-consuming functions, allowing for fast updates.

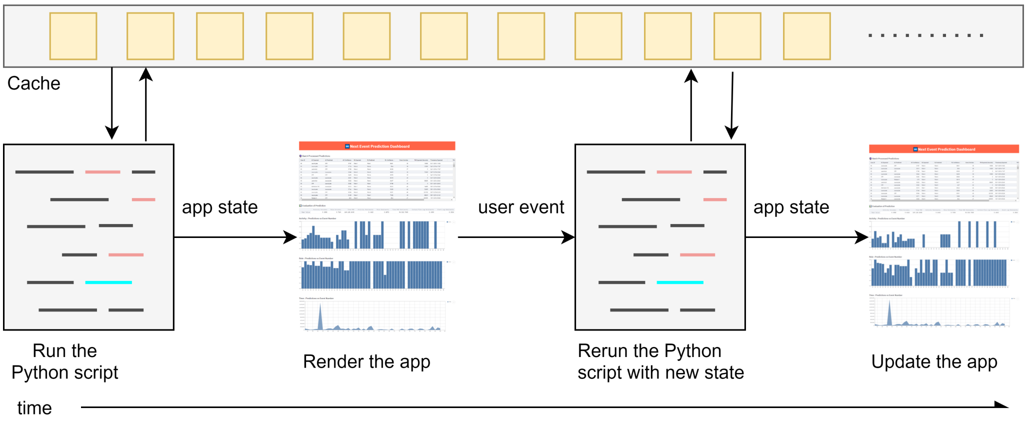


Figure 4.6: Events triggered by User on Streamlit, which causes re-run

4.5 Dashboard Features

Figure 4.7 shows the flow of the Next Event Prediction Dashboard. Under this, Single Event Processing and Batch Processing are two main views, and their respective control element widgets in the App Control Menu are shown on the left and right side. Execution, Evaluation, and What-If Modes are components that make up the three different perspectives of Single Event Processing.

Execution mode covers the Requirement 4a, it enables users to make choices based on the information shown on the dashboard for an ongoing case. The Requirement 4b is applied in Evaluation Mode, where it replays the previously executed cases one by one, such that it could be replayed by changing the prefix length back and forth. It is meant to provide an outline to the user on how an event will turn out if the recommendations were only followed from the model predictive recommendations, which may lead to the reliability a user can put on the model recommendations before deciding on a similar case. What-If mode covers Requirement 4c allows the user to select between different predicted recommendations for each event for a case, and execute it by taking it as a prefix to the model and to see other possible outcomes because of the selected recommendation.

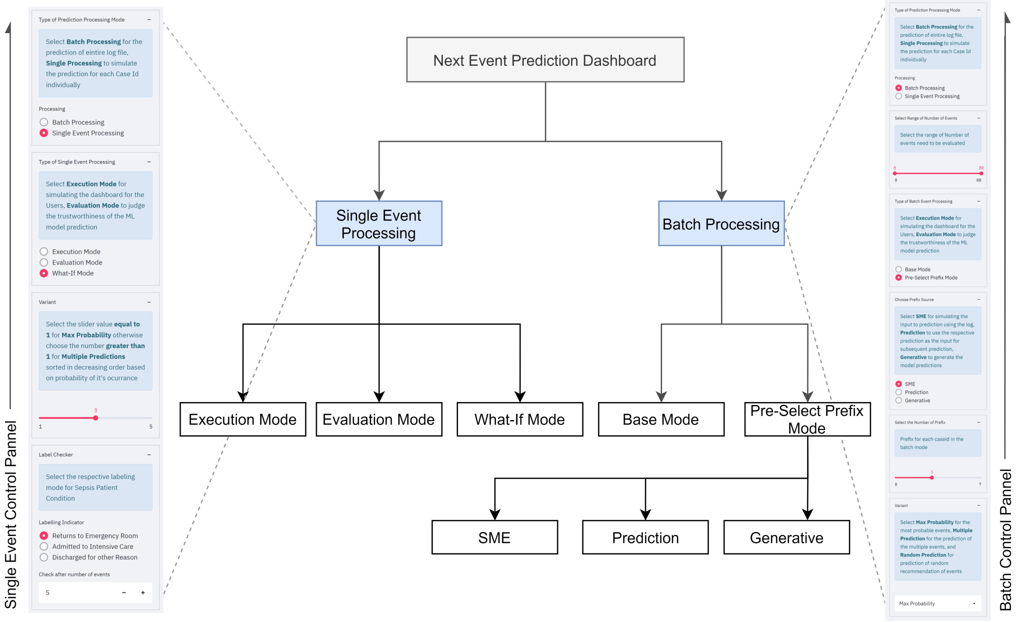
Similarly, Batch Processing has two different modes: Base Mode and Pre-Select Prefix Mode, which covers the Requirement 3a. Pre-Select Prefix Mode comprises three options, namely SME, Prediction and Generative. Both the mode uses the Control-Flow Log Similarity (CFLS) using Damerau-Levenshtein distance, Mean Absolute Error (MAE) of cycle times and Event Log Similarity (ELS) which covers the Requirement 3b.

The Pre-select prefix would let each case of the test event log be assigned with the same number of prefixes which is in the log. So, for the case having the activity $< A,B,C,D,E >$, the pre-select prefix of length three would lead to $< A,B,C >$, similarly it is applied to all the cases of the test log while evaluation. In other words, if the trace length for the case is n, with the prefix of length l, which can be given to the predictive model, is adjustable in the range of $0 \leq l < n$. The selected prefix is termed a pre-select prefix. The pre-select prefix is only supported by Pre-Select Prefix Mode in Batch processing and also in the Evaluation mode in Single Event Processing with a modification in the range of prefix selection as $1 \leq l < n$.

4.6 Control Menu Options

The dashboard offers many control options to manage different modes of Requirement 3 and 4. Some of them are common across different modes of each view, i.e., the Single event processing and the Batch processing, others are specific for certain modes only. All the control functionalities are discussed below.

- **Types of Prediction Processing**: It helps to toggle between two main views for the Next Activity prediction dashboard, which is the Batch and Single event processing.



Type of Prediction Processing Mode –

Select **Batch Processing** for the prediction of eintire log file, **Single Processing** to simulate the prediction for each Case Id individually

Processing

○ Batch Processing
● Single Event Processing

Type of Single Event Processing –

Select **Execution Mode** for simulating the dashboard for the Users, **Evaluation Mode** to judge the trustworthiness of the ML model prediction

○ Execution Mode
○ Evaluation Mode
● What-If Mode

Variant –

Select the slider value **equal to 1** for **Max Probability** otherwise choose the number **greater than 1** for **Multiple Predictions** sorted in decreasing order based on probability of it's ocurrance

3
1 5

Label Checker –

Select the respective labeling mode for Sepsis Patient Condition

Labelling Indicator

● Returns to Emergency Room
○ Admitted to Intensive Care
○ Discharged for other Reason

Check after number of events

5 – +

Type of Prediction Processing Mode –

Select **Batch Processing** for the prediction of eintire log file, **Single Processing** to simulate the prediction for each Case Id individually

Processing

● Batch Processing
○ Single Event Processing

Select Range of Number of Events –

Select the range of Number of events need to be evaluated

8 88
8 88

Type of Batch Event Processing –

Select **Execution Mode** for simulating the dashboard for the Users, **Evaluation Mode** to judge the trustworthiness of the ML model prediction

○ Base Mode
● Pre-Select Prefix Mode

Choose Prefix Source –

Select **SME** for simulating the input to prediction using the log, **Prediction** to use the respective prediction as the input for subsequent prediction, **Generative** to generate the model predictions

● SME
○ Prediction
○ Generative

Select the Number of Prefix –

Prefix for each caseid in the batch mode

3
0 7

Variant –

Select **Max Probability** for the most probable events, **Multiple Prediction** for the prediction of the multiple events, and **Random Prediction** for prediction of random recommendation of events

Max Probability

Figure 4.7: Next Event Prediction Dashboard Navigation: Single Event Processing and Batch Processing

- The Batch Processing has the following control options :
  - **Select Range of Number of Events**: It is used to slide between range of cases of different event lengths under batch view.
  - **Type of Batch Event Processing**: It enables to choose between the different modes in batch processing, that is, Base Mode and Pre-Select prefix mode.
  - **Choose the Prefix Source**: Pre-select prefixes have different options on how the predictive recommendation will happen, which varies based on how the prefixes are being built before feed back to the model for next prediction. It provides selection between the options SME, Prediction, and Generative.
  - **Select Number of Prefix**: This option, under the Pre-Select prefix mode, allows the selection of the number of pre-selected prefixes that it takes from the executed log for the predictions.
  - **Variant**: It allows the selection of the number of predictions to be made, it is a dropdown option aided with the slider as the sub option named `Number of Predictions`. It lets the user select the number of predictions to be made in case multi prediction option has been chosen under the Variant option.

- The Single Event Processing has the following control options :
  - **Types of Single Event Processing**: It permits selection between different modes available in the Single event processing.
  - **Select Case ID** : It provides choice between different case id's.
  - **Choose [Activity, User, Time]**: This option, under the Evaluation mode of Single Event Processing, allows the user to select back and forth the number of the prefixes to be provided to the model.
  - **What-IF Prediction Choose Box**: The option is present under the What-If mode of Single Event Processing, and it enables the selection of prediction to be used as a prefix for the next prediction, either externally or from the recommended actions.
  - **Variant**: It allows the selection of the number of predictive recommendations to be made, and it is controlled by a slider. Slider at value 1 means maxima, and greater than 1 means multiple predictive recommendations sorted in the order of confidence (probability of it to happen).
  - **Label Checker**: It is customizable for individual event logs, and corresponding checks have been provided to keep a tab on ongoing predictions. It lets to set after how many events the check should start checking for the deviant or regular label in the predictions made. It has been further discussed in the subsection 4.6.1, and this control covers the Requirement 4d.

The main control options discussed above have been summarized in Table 4.1 about its control widget type and the availability in the corresponding views.

4.6.1 Label Checker

The label checker has been built outside the blackened part, which means the LSTM model doesn't have to be trained for it. So, it adds the advantage of being customized without re-training the model. The user can decide during runtime (while a case is being replayed) to evaluate which predictive recommendations, will result in deviant or regular behaviour and categorize them once a specific number of events have occurred. Since, from the beginning of the recommendations, it is unlikely to determine if the recommended actions going to belong to which category. Thus, providing the number of events after which the check will start determining the label is essential.

The runtime labelling for the Sepsis Cases is based on Teinemaa et al. [8] which can be considered as KPI's for the decision-making :

Table 4.1: Main Control Options, and its availability in the respective View's

Control Name	Widget Type	Control View Availability	
		Single Event Processing	Batch Event Processing
Type of Prediction Processing Mode	Radio Button	✓	✓
Type of Single Event Processing	Radio Button	✓	-
Type of Batch Event Processing	Radio Button	-	✓
Select Range of Number of Events	Range Slider	-	✓
Select Number of Prefix	Slider	-	✓
Variant	Slider/ Select Dropdown	✓	✓
Label Checker	Radio Button	✓	-

- The patient is re-admitted to the emergency department (Return ER).
- The patient is admitted to intensive care (Admission IC).
- The patient is released from the hospital for a Reason besides Release A.

The Algorithm 2 is according to the decision-making KPI's of Sepsis Cases at runtime to check the label for the condition the user would set before executing a case from the dashboard in the Single Event Mode. Activity to be classified in algorithm are Return ER, Admission IC, Release A and List of activity executed are the activities happened so far for the case. When the executed trace is met with the condition it is set to deviant, otherwise it is set to regular.

4.7 Summary

In this chapter, the dashboard design has been discussed comprehensively according to the functional requirements. Based on the requirements, the dashboard offers two main views, Single Event Processing and Batch processing. The navigation to different feature's (modes) has been discussed which are under each view and

Algorithm 2 Classify Label : Deviant and Regular Behaviour

```
Require: $l \leftarrow$ Length of events after which the check should start
Require: $c \leftarrow$ Activity to be classified
Require: $ac \leftarrow$ List of activity executed
Ensure: $decide =$ ‘’
  if $l \geq length(ac)$ then then
      if $c$ in $ac$ then then
          $decide \leftarrow$ Deviant
      else if $c$ not in $ac$ then then
          $decide \leftarrow$ Regular
      end if
  else
      $decide \leftarrow$ Not Decided
  end if
```

briefly introduced what they are used for. Each view control options have been discussed, and also the labelling of Sepsis Cases in deviant and regular at run time in the Single Event Processing. The offered feature’s design, and its application, will be discussed further in details in the upcoming chapters.

Chapter 5

Dashboard Design of Single Event Processing and Batch Processing

Up until now, we have addressed the backend LSTM model of the system and the dashboard architecture, and the backend predictive model of the system. In this chapter, we will focus on how Single Event Processing is designed and how the Batch Processing design is related to it. Thus, the following research question will be addressed :

Research Question : How should the dashboard be designed to incorporate the LSTM models to simulate the predictions generated for the users?

The first three sections of the chapter is divided into explaining the different mode designs of Single Event Processing — Section 5.1 discusses the Execution mode design, Section 5.2 discusses the Evaluation mode design, and Section 5.3 explains the design of What-If mode. Lastly, Section 5.4 discusses the Batch Processing design and the connection with different modes of Single Event processing.

5.1 Execution Mode

Execution mode functionality revolves around the idea of assisting domain experts to decide the next action to be executed for the upcoming event stream. In this section, we will first look to understand the design of the mode and subsequently look into the user interface (UI).

5.1.1 Execution Mode Design

The prediction recommendations provided by the model is displayed on the dashboard along with their respective confidences. The confidence is measured in terms of probability of its occurrence for a given prefix. The dashboard also consists of attributes information associated with the event which helps to understand the state of the process.

The mode also offers multi-prediction and the number of recommendations is displayed based on the user's choice, which is inturn capped based on the logic mentioned in Section 4.2. It sorts the recommendation actions in descending order of model confidence and groups the activities and roles accordingly.

Since, the model is capable of generating events with zero-prefix, it is used as the starting point for all the cases to recommend the first action. Subsequent actions are recommended based on feedback from the event stream in the form of executed events, which acts as the prefix for the model to predict the recommendations for the next action. In our design for each step, generated recommendations are again fed to the model as feedback to predict one-step ahead of each recommendation, to give an early idea to the user what is most likely to happen if a particular predictive recommendation is chosen. This is done in two ways:

5.1.1.1 One-Step ahead Process Outcome

In this method, the predicted recommendations are appended with the incoming prefix from the event stream individually and given to the predictive model to generate the most probable one-step ahead action. So, for a given prefix $hd^l =< e_1, e_2, ..., e_l >$ of a case-id which is of length l, the multiple next predictive recommendation actions predicted by the predictive model be $< e_{p_{1l}} >, < e_{p_{2l}} >, ..., < e_{p_{jl}} >$ ordered by confidence of the model. Then, for each one-step ahead prediction, prefixes will be $< e_1, e_2, ..., e_l, e_{p_{1l}} >, < e_1, e_2, ..., e_l, e_{p_{2l}} >, ..., < e_1, e_2, ..., e_l, e_{p_{jl}} >$ each of length $l+1$ and would predict one step ahead prediction $< e_{op_1} >, < e_{op_2} >, ..., < e_{op_j} >$ as shown in Figure 5.1. Here j is the number of multi predictions.

One-Step ahead Process Outcome explains the most likely outcome that can happen in the future because it combines the recommendation with the past actions that have been taken so far. This might give the user an idea on how things might turn out next.

5.1.1.2 One-Step ahead Generative Outcome

This technique only utilises continuous prefix feedback of the model recommendations individually until the end of case. In other words, each of the individual recommendations thus far are given as the prefix feedback to the predictive model to generate the most probable one-step ahead prediction. Formally, as shown in Figure 5.2 for a given case-id, a prefix $hd^l =< e_1, e_2, ..., e_l >$ of length l, the multiple next predictive recommendation actions predicted by the predictive model be $< e_{p_{1l}} >, < e_{p_{2l}} >, ..., < e_{p_{jl}} >$ ordered by confidence of the model. For

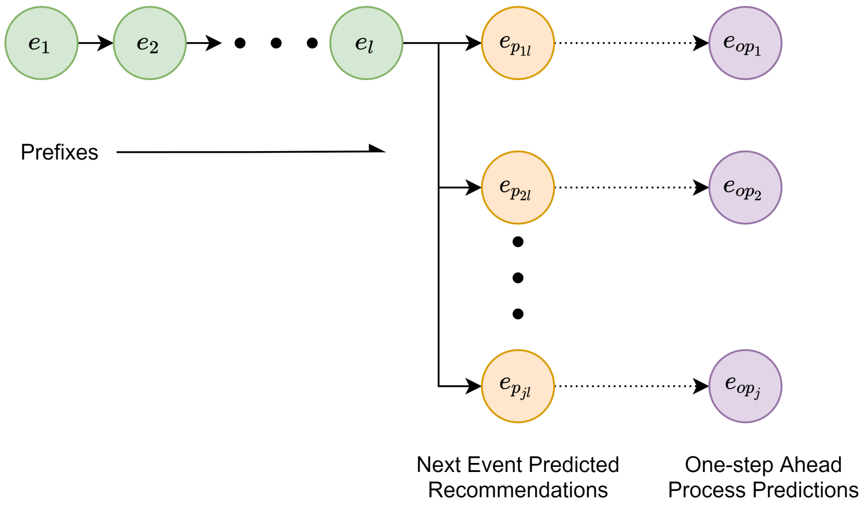


Figure 5.1: One-step ahead Process prediction

each prefix thus far, i.e., for $< e_1 >$ next multi-predictive recommendations are $< e_{p_{11}} >, < e_{p_{21}} >, ..., < e_{p_{j1}} >$, for prefixes, $< e_1, e_2 >$ next multi-predictive recommendations are $< e_{p_{21}} >, < e_{p_{22}} >, ..., < e_{p_{j2}} >$, and for prefix, $< e_1, e_2, ..., e_l >$ next multi-predictive recommendations are $< e_{p_{1l}} >, < e_{p_{2l}} >, ..., < e_{p_{jl}} >$. Then for each one step ahead, prediction of each prefix will be $< e_{11}, e_{12}, ..., e_{1l} >, < e_{21}, e_{22}, ..., e_{2l} >, ..., < e_{j1}, e_{j2}, ..., e_{jl} >$ and this would predict one step ahead prediction as $< e_{og_1} >, < e_{og_2} >, ..., < e_{og_j} >$.

One-Step ahead Generative Outcome is about providing an extra check to the user about the model's one-step ahead prediction and compare it with the One-Step ahead Process Outcome. In case of both process and generative predicting the same action, this indicates that there is a high probability for that action to be occuring one-step ahead otherwise there is always a chance to have the next action being more close to process prediction. This might provide the user with more confidence before choosing the recommendation.

5.1.2 Execution Mode User Interface

Figure 5.3 and 5.4 combined is the one-page developed UI of execution mode for top two predictive recommendations. In Figure 5.3 the Case-Id is a dropped down select box, followed by it shows the state of the process which is the contextual information, and the process history which shows the completed events so far. The other half of the figure has the predictive recommendations and the time duration it would take to complete the recommend action. The `1st Predcition` indicates the model's highest confident recommendation, and the `2nd Predcition` indicates the model's second-highest confident recommendation for both Activity and Role.

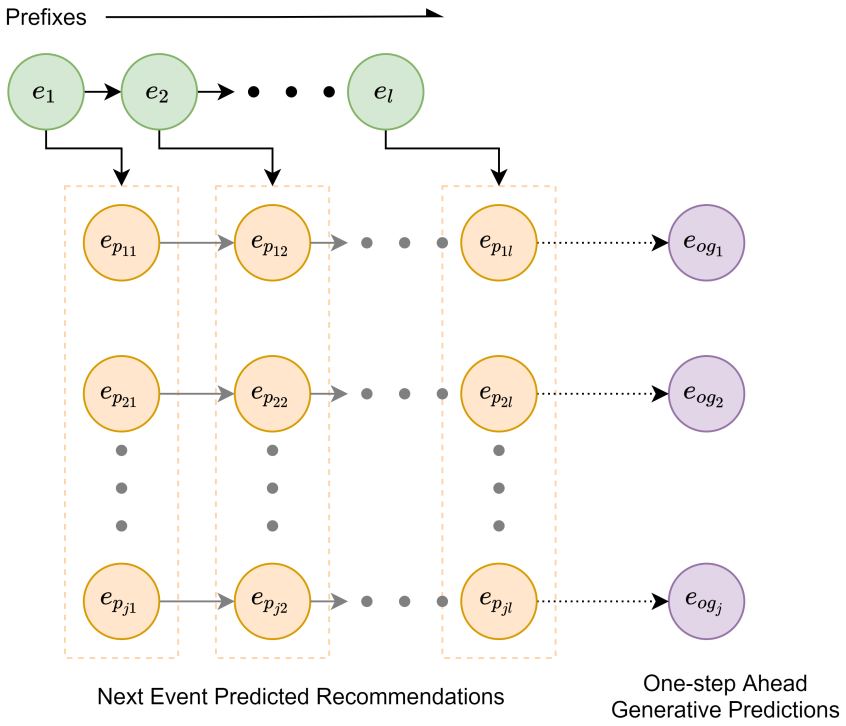


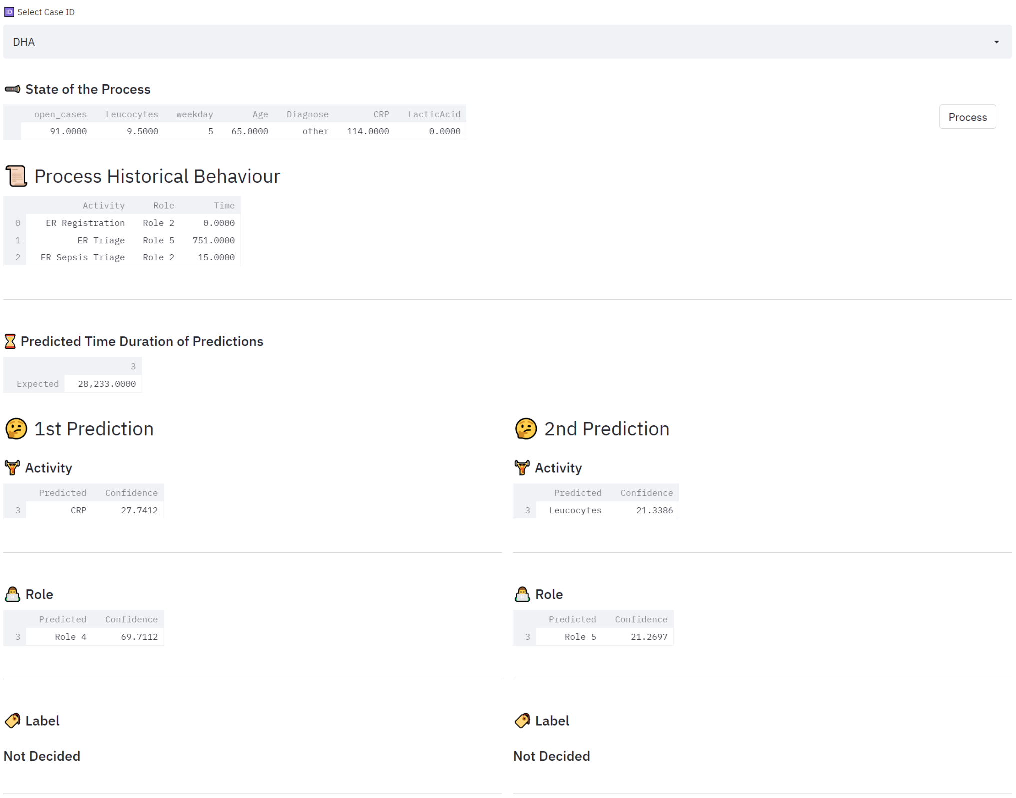
Figure 5.2: One-step ahead Process prediction

Label indicates the regular and deviant behaviour as discussed in subsection 4.6.1.

Figure 5.4 shows the One-Step ahead predictions. The left side of the figure illustrates the One-Step ahead Process Outcome (Subsection 5.1.1.1) while the right side of the figure has the One-Step ahead Generative Outcome (Subsection 5.1.1.2).

5.2 Evaluation Mode

The evaluation mode capability is designed to aid domain experts with determining the trust on the predictive model and learning the explainability of process prediction [9]. Essentially, it allows the user to change prefixes back and forth on the previously executed cases where, the selected prefix is called pre-select prefix. The change in prefix enables to observe how the model performs for each selected pre-select prefix. It is intended to provide an outline to the user on how an event will turn out if only the predictive recommendations from the model is followed, which may lead to embed a level of cognitive confidence and risk assessment in the user about the model recommendations before making a decision on a comparable case.



Select Case ID

DHA

State of the Process

open_cases	Leucocytes	weekday	Age	Diagnose	CRP	LacticAcid
91.0000	9.5000	5	65.0000	other	114.0000	0.0000

Process Historical Behaviour

	Activity	Role	Time
0	ER Registration	Role 2	0.0000
1	ER Triage	Role 5	751.0000
2	ER Sepsis Triage	Role 2	15.0000

Predicted Time Duration of Predictions

	3
Expected	28,233.0000

1st Prediction

Activity

	Predicted	Confidence
3	CRP	27.7412

Role

	Predicted	Confidence
3	Role 4	69.7112

Label

Not Decided

2nd Prediction

Activity

	Predicted	Confidence
3	Leucocytes	21.3386

Role

	Predicted	Confidence
3	Role 5	21.2697

Label

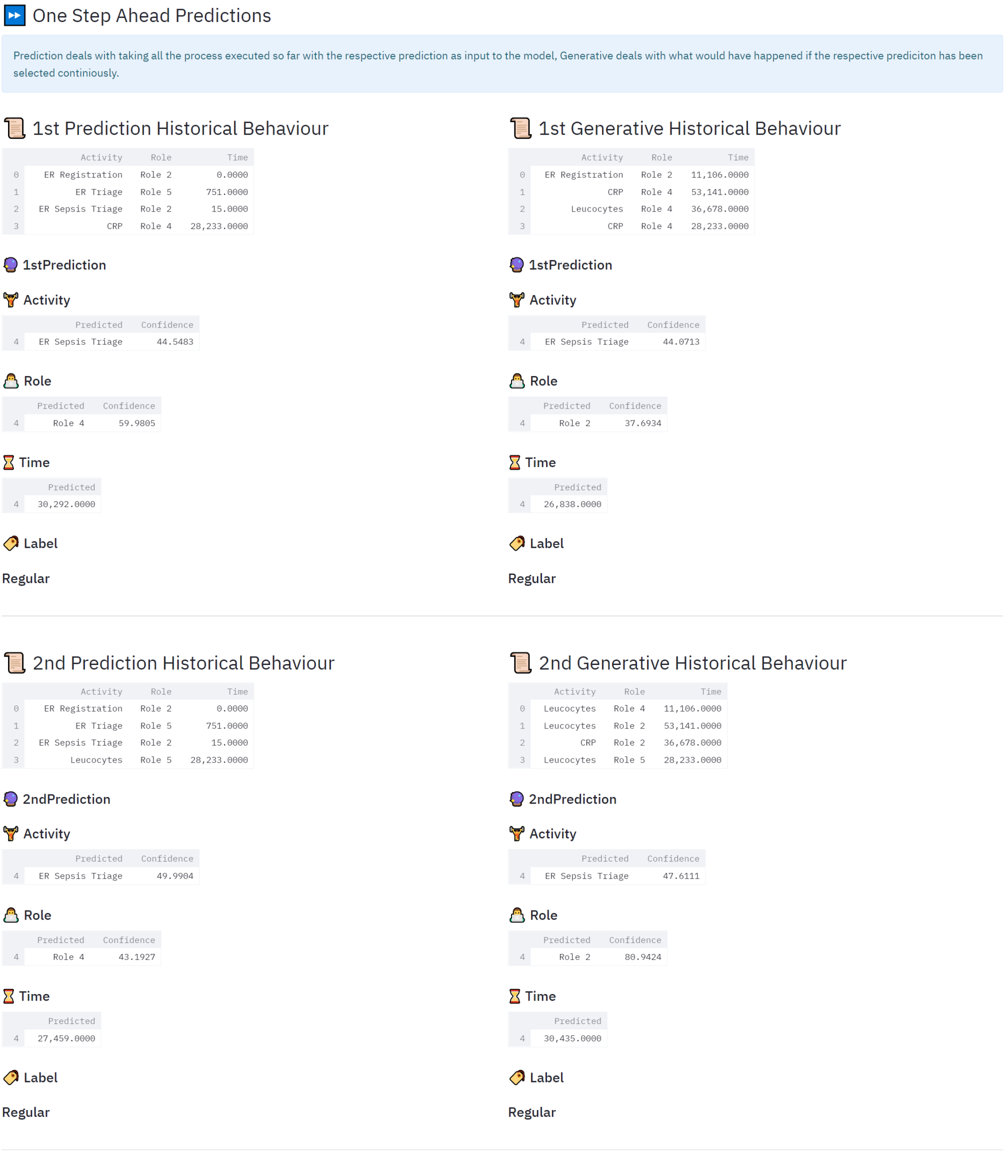
Not Decided

Figure 5.3: Execution Mode UI — Part I

5.2.1 Evaluation Mode Design

The mode provides prediction in two broad categories — The first, is based on just the prefix as it has actually been executed. The prefixes are fed to the model from the test event log and it shows the subsequent predictions until the end of the case. *On the dashboard, it is termed as SME prediction and will be referring to it with the same term in the chapter*. The second is based on generative emulation of events after selecting the pre-select prefix from the dashboard by repeatedly executing the LSTM model until the end of the case. The completion of the trace is compared with the observed behaviour in the log and from the maxima of SME prediction.

Conformance check is performed on the prefix selected to evaluate which predictive recommendation of the model has learnt it. If the model can reproduce similar actual prefixes, then the corresponding predictive recommendation knows to reproduce it based on the context information. If the predictive recommendation differs from the actual prefix, then LSTM did not learn it. Lastly, if the model is able to reproduce the actual prefixes with the help of more than one predictive



One Step Ahead Predictions

Prediction deals with taking all the process executed so far with the respective prediction as input to the model, Generative deals with what would have happened if the respective prediciton has been selected continiously.

1st Prediction Historical Behaviour

	Activity	Role	Time
0	ER Registration	Role 2	0.0000
1	ER Triage	Role 5	751.0000
2	ER Sepsis Triage	Role 2	15.0000
3	CRP	Role 4	28,233.0000

1stPrediction

Activity

	Predicted	Confidence
4	ER Sepsis Triage	44.5483

Role

	Predicted	Confidence
4	Role 4	59.9805

Time

	Predicted
4	30,292.0000

Label

Regular

1st Generative Historical Behaviour

	Activity	Role	Time
0	ER Registration	Role 2	11,106.0000
1	CRP	Role 4	53,141.0000
2	Leucocytes	Role 4	36,678.0000
3	CRP	Role 4	28,233.0000

1stPrediction

Activity

	Predicted	Confidence
4	ER Sepsis Triage	44.0713

Role

	Predicted	Confidence
4	Role 2	37.6934

Time

	Predicted
4	26,838.0000

Label

Regular

2nd Prediction Historical Behaviour

	Activity	Role	Time
0	ER Registration	Role 2	0.0000
1	ER Triage	Role 5	751.0000
2	ER Sepsis Triage	Role 2	15.0000
3	Leucocytes	Role 5	28,233.0000

2ndPrediction

Activity

	Predicted	Confidence
4	ER Sepsis Triage	49.9904

Role

	Predicted	Confidence
4	Role 4	43.1927

Time

	Predicted
4	27,459.0000

Label

Regular

2nd Generative Historical Behaviour

	Activity	Role	Time
0	Leucocytes	Role 4	11,106.0000
1	Leucocytes	Role 2	53,141.0000
2	CRP	Role 2	36,678.0000
3	Leucocytes	Role 5	28,233.0000

2ndPrediction

Activity

	Predicted	Confidence
4	ER Sepsis Triage	47.6111

Role

	Predicted	Confidence
4	Role 2	80.9424

Time

	Predicted
4	30,435.0000

Label

Regular

Figure 5.4: Execution Mode UI — Part II

recommendation, then the LSTM model knows it but it is able to convey it using multi-predictive recommendation. This will cause the user to have the perception about the model predictive recommendation behaviour and the risk associated with different predictive recommendations. The risk could be calculated based on until how many events the user can rely on individual predictive recommendations before switching to a different predictive recommendation. The following subsections will explain the design of SME predictions, Generative emulation and the conformance check.

5.2.1.1 End of Trace — SME

This approach uses the continuous prefixes from the test log one by one to predict the next event until the end of case. The first predicted event $< e_{sp_l} >$ is due to the pre-selected prefixes $hd^l =< e_1, e_2, ..., e_l >$. The subsequent predictions are the result of what was observed in the event log. For example, the prefix $hd^{l+1} =< e_1, e_2, ..., e_l, e_{l+1} >$ will allow the predictive model to predict $e_{sp_{l+1}}$ and this continues until hd^{n-l} prefix as shown in the Figure 5.5.

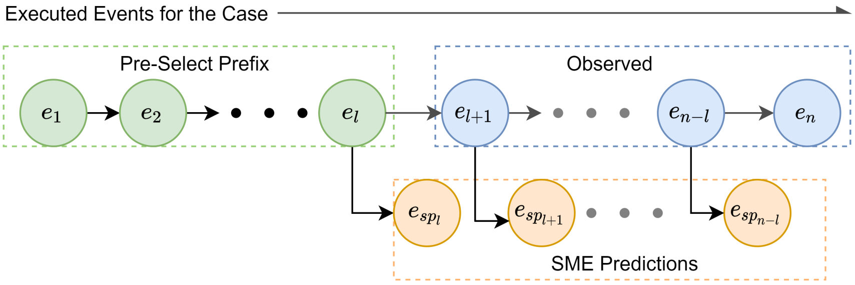


Figure 5.5: End of Process Prediction — SME

5.2.1.2 End of Trace — Generative

In this approach, the pre-select prefix and the independent predictive recommendations generated are executed on the LSTM model until the end of the case. Each of the individual predictive recommendations derived from the predictive model uses the pre-select prefix and appends the individual predictive recommendations to recommend the next event. As illustrated in Figure 5.6, using the pre-select prefix $hd^l =< e_1, e_2, ..., e_l >$, the model's predictive recommendations are $< e_{gp_{1l}} >, < e_{gp_{2l}} >, ..., e_{gp_{jl}}$. Next, the individual predictive recommendations along with pre-select prefix form j independent prefixes and fed it to the model individually. So, $< e_1, e_2, ..., e_l, e_{gp_{1l}} >$ will predict $< e_{gp_{1(l+1)}} >$, similarly $< e_1, e_2, ..., e_l, e_{gp_{2l}} >$ will predict $< e_{gp_{2(l+1)}} >$ and $< e_1, e_2, ..., e_l, e_{gp_{jl}} >$ will predict $< e_{gp_{j(l+1)}} >$ and it goes on until $n-l$, where n is the trace length of the executed case in the event log.

This method follows the generative approach of predictions. Unlike what we saw in End of Trace — SME (Section 5.2.1.1), it does not use the observed events for the next predictive recommendations rather, it forms individual prefixes to generate the next predictions as illustrated in Figure 5.6. The idea behind this approach is, if provided with a pre-select prefix, how will the case end if the user has only followed the respective predictive recommendations afterwards.

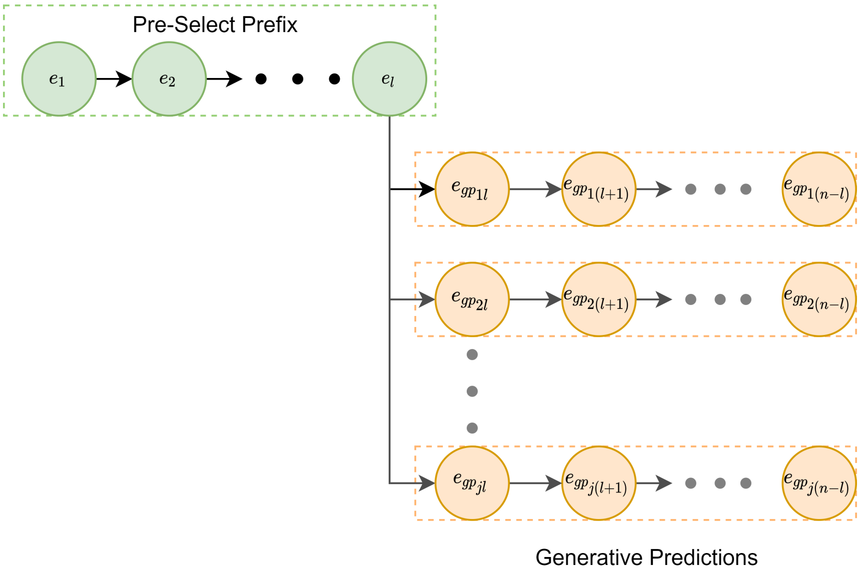


Figure 5.6: End of Process Prediction — Generative

5.2.1.3 Risk Analysis — Conformance Check

The risk analysis of the predictive model is performed to estimate how much the model can emulate the given pre-select prefix and for which of the predictive recommendations. The emulation starts with zero-prefix for the first predictive recommendation and continues to append the pre-select prefix until e_{l-1}, where l is the length of the pre-selected prefix. As shown in Figure 5.7, for each appended pre-selected prefix, the predictive recommendations are generated individually, which can be compared with the actual prefix.

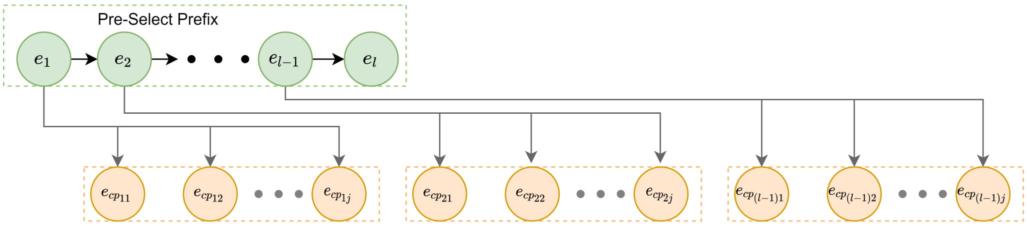
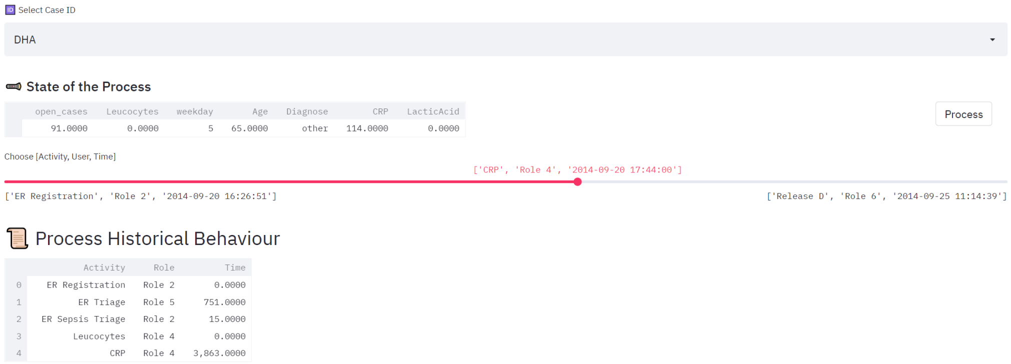


Figure 5.7: Conformance Check

5.2.2 Evaluation Mode User Interface

Figure 5.8, 5.9 and 5.10 combined is the developed UI of evaluation mode for top two predictive recommendations. In the Figure 5.8 Select Case ID permits to choose cases from the dropdown of select box following which, the state of the process is tabulated. Choose[Activity, User, Time] allows to slide back and forth to select the prefix of the case, and the process historical behaviour tabulates those selected prefixes.



	open_cases	Leucocytes	weekday	Age	Diagnose	CRP	LacticAcid
	91.0000	0.0000	5	65.0000	other	114.0000	0.0000

Choose [Activity, User, Time]

['CRP', 'Role 4', '2014-09-20 17:44:00']

['ER Registration', 'Role 2', '2014-09-20 16:26:51']

['Release D', 'Role 6', '2014-09-25 11:14:39']

	Activity	Role	Time
0	ER Registration	Role 2	0.0000
1	ER Triage	Role 5	751.0000
2	ER Sepsis Triage	Role 2	15.0000
3	Leucocytes	Role 4	0.0000
4	CRP	Role 4	3,863.0000

Figure 5.8: Evaluation Mode UI — Part I

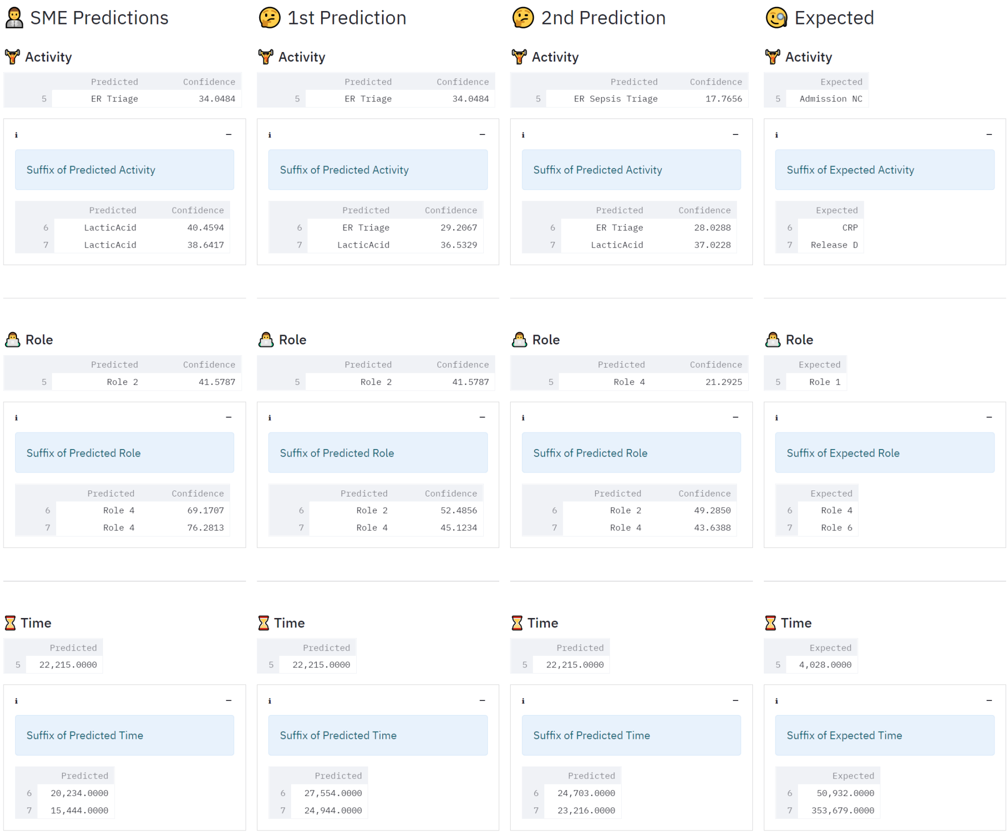
The left side of the Figure 5.9 shows SME Prediction which is based on End of Trace — SME 5.2.1.1, then `1st Predcition` and `2nd Predcition` are the predictive recommendations which is in accordance with End of Trace — Generative 5.2.1.2. The right side of the figure shows the Expected events which is in the log.

Figure 5.10 shows the Conformance Check which is developed based on the design of Risk Analysis — Conformance Check (Subsection 5.2.1.3) and tabulates the respective predictive recommendations' ability to reproduce the prefix selected using the Choose[Activity, User, Time].

5.3 What-If Mode

So far, we looked into different ways of generative predicted recommendations which the model has learnt from historical behaviour in a specific order. These predictions are constrained because the prefixes are grouped from a high to low probability (model confidence) for each event. This means that predictive recommendations are serialized individually in the group of the model confidence they belong to and their respective predictions is appended back to the same group to be used as the prefix at runtime. This is what we saw in previous two sections as well. As there is no external feedback mechanism to choose among these predictions, continuous prefixes are considered when recommending the next action.

In the classical process mining What-If technique, it first requires process discovery, then defining the simulation scenario and lastly running the simulation.



SME Predictions

Activity

	Predicted	Confidence
5	ER Triage	34.0484

i −

Suffix of Predicted Activity

	Predicted	Confidence
6	LacticAcid	40.4594
7	LacticAcid	38.6417

Role

	Predicted	Confidence
5	Role 2	41.5787

i −

Suffix of Predicted Role

	Predicted	Confidence
6	Role 4	69.1707
7	Role 4	76.2813

Time

	Predicted
5	22,215.0000

i −

Suffix of Predicted Time

	Predicted
6	20,234.0000
7	15,444.0000

1st Prediction

Activity

	Predicted	Confidence
5	ER Triage	34.0484

i −

Suffix of Predicted Activity

	Predicted	Confidence
6	ER Triage	29.2067
7	LacticAcid	36.5329

Role

	Predicted	Confidence
5	Role 2	41.5787

i −

Suffix of Predicted Role

	Predicted	Confidence
6	Role 2	52.4856
7	Role 4	45.1234

Time

	Predicted
5	22,215.0000

i −

Suffix of Predicted Time

	Predicted
6	27,554.0000
7	24,944.0000

2nd Prediction

Activity

	Predicted	Confidence
5	ER Sepsis Triage	17.7656

i −

Suffix of Predicted Activity

	Predicted	Confidence
6	ER Triage	28.0288
7	LacticAcid	37.0228

Role

	Predicted	Confidence
5	Role 4	21.2925

i −

Suffix of Predicted Role

	Predicted	Confidence
6	Role 2	49.2850
7	Role 4	43.6388

Time

	Predicted
5	22,215.0000

i −

Suffix of Predicted Time

	Predicted
6	24,703.0000
7	23,216.0000

Expected

Activity

	Expected
5	Admission NC

i −

Suffix of Expected Activity

	Expected
6	CRP
7	Release D

Role

	Expected
5	Role 1

i −

Suffix of Expected Role

	Expected
6	Role 4
7	Role 6

Time

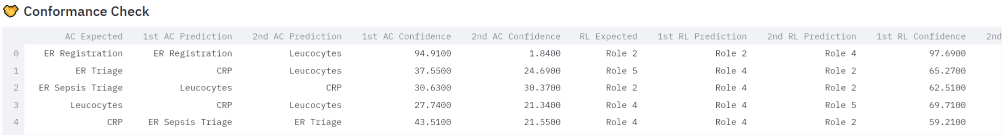
	Expected
5	4,028.0000

i −

Suffix of Expected Time

	Expected
6	50,932.0000
7	353,679.0000

Figure 5.9: Evaluation Mode UI — Part II



Conformance Check

	AC Expected	1st AC Prediction	2nd AC Prediction	1st AC Confidence	2nd AC Confidence	RL Expected	1st RL Prediction	2nd RL Prediction	1st RL Confidence	2nd
0	ER Registration	ER Registration	Leucocytes	94.9100	1.8400	Role 2	Role 2	Role 4	97.6900	
1	ER Triage	CRP	Leucocytes	37.5500	24.6900	Role 5	Role 4	Role 2	65.2700	
2	ER Sepsis Triage	Leucocytes	CRP	30.6300	30.3700	Role 2	Role 4	Role 2	62.5100	
3	Leucocytes	CRP	Leucocytes	27.7400	21.3400	Role 4	Role 4	Role 5	69.7100	
4	CRP	ER Sepsis Triage	ER Triage	43.5100	21.5500	Role 4	Role 4	Role 2	59.2100	

Figure 5.10: Evaluation Mode UI — Part III

Defining simulation scenario is limited to checking the performance concerning time, cost, and number of resources [35]. This means that it does not address the possibilities of viable and actionable options when a certain event occurs, as it strictly follows the process model found in process discovery.

The What-If mode in our solution addresses the above-mentioned issues by allowing the user to choose from a variety of actions from the predicted recommendations for each event, execute them on the predictive model as a prefix, and discern other possible outcomes because of the selected recommendations at runtime. The framework of the system is designed such that this mode can be executed on an ongoing case as well as on an executed case in the test event log as simulation. When it is applied on an ongoing case, it enables the user to undertake Artificial Intelligence (AI) actions to achieve the result, depending on user knowledge about the business process.

5.3.1 What-If Mode Design

The design for the ongoing case and for the simulation is similar. In case of ongoing case,this mode assumes that there is an external mechanism to take actions whereas in the case of simulation, it takes it from the log. Thus, it offers to either select the actions which are not being recommended by the model or to choose the actions being recommended by the model from the dashboard. To choose the action externally or log which need to be incorporated for further recommendations, the dashboard offers an option with the name of SME apart from the other predictive recommendations.

Figure 5.11 shows the working of the choices made from recommended predictions. The process starts with zero-prefix (e_1) and provides recommendations ($< e_{p11} >, < e_{p12} >, ... < e_{pj1} >$) to the user based on the number of recommendations (j) selected. The user then selects any one recommendation which they consider is apt based on the state of the process, denoted as, $e_{p_{21}}$ which becomes the prefix for the next event prediction. In case of simulation, they can replay and try-out various actions. The chosen recommendation is then appended to the current prefix of length l, resulting in a prefix of length $l+1$. The new prefix is again given to the prediction model to predict the next possible recommendations from which the user chooses a suitable recommendation. The option subsequently becomes the appended prefix to the predictive model, and this goes on until the recommended action converge towards the “end” activity. In case of Sepsis event log, the release activities, i.e., Release A, Release B, Release C, Release D are considered the “end” activity.

5.3.2 What-If Mode User Interface

Figure 5.12 shows the user interface of What-If. At the top, it permits to choose the case-id, followed by the state of the process. Within the What-If Prediction Choose Box, it allows to choose among recommendations from the screen. SME option is for external action for an ongoing process and in case of simulation processing the event from the event log. Then `1st Predcition`, `2nd Predcition` and `3rd Predcition` are the predictive recommendations from the model.

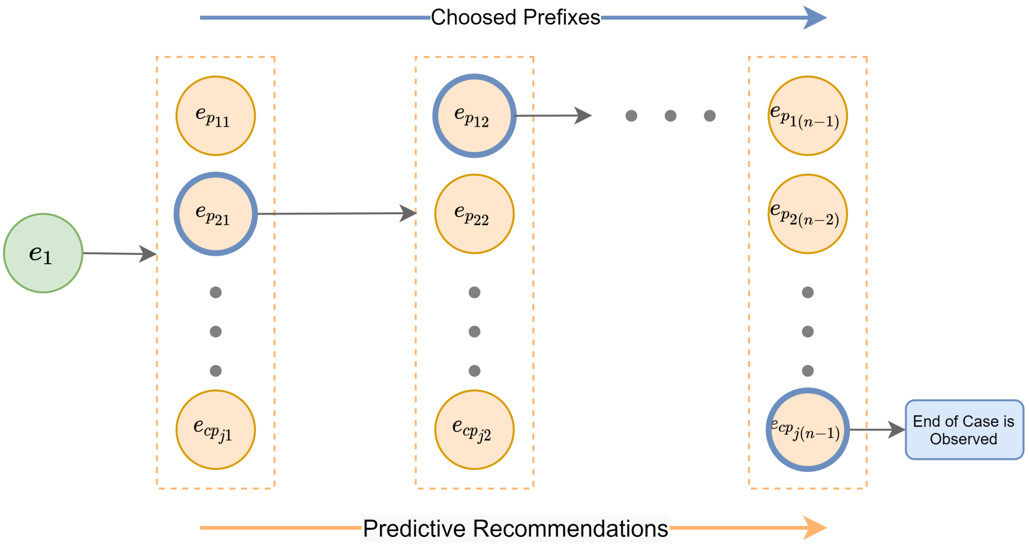


Figure 5.11: Working of What-If

5.4 Batch Processing

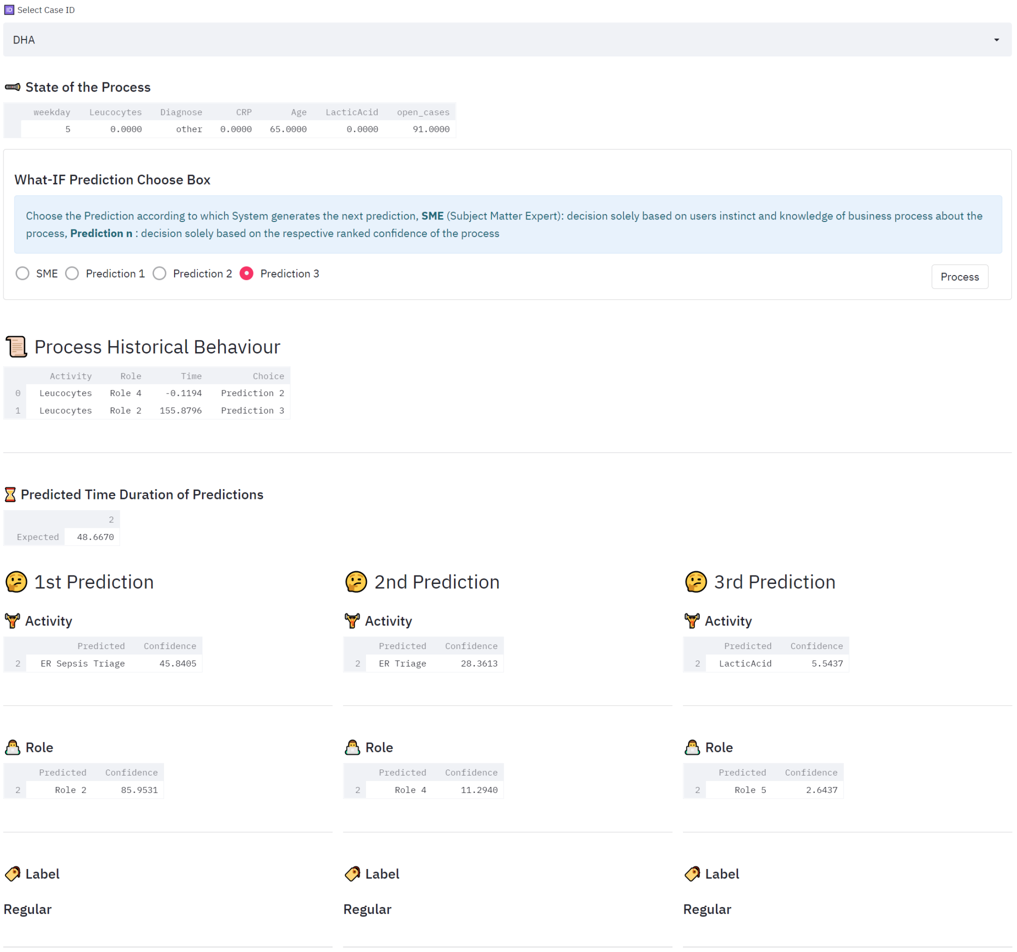
Batch Mode simulates the prediction for the entire event log in one go. It is used to quantitatively evaluate the model predictive recommendations capabilities on a test event log level, and the trust one can have in it for a given log. The relation of both the modes and its options has been discussed in subsection 5.4.1 and 5.4.2. The subsection 5.4.3 discuss the Batch Processing User interface.

5.4.1 Base Mode

Base mode enables the evaluation of next event prediction using the test event log as its prefix. It only takes the prefix input from the test event log and generates the next event. It imitates the Execution Mode (Section 5.1.1) on a log level.

5.4.2 Pre-Selected Prefix Mode

In this mode, the prediction evaluation allows the pre-select prefix for each case taken from the test event log. It offers three predictive options, which varies depending on where and how the prefix would be chosen to recommend the next event.



Select Case ID

DHA

State of the Process

	weekday	Leucocytes	Diagnose	CRP	Age	LacticAcid	open_cases
	5	0.0000	other	0.0000	65.0000	0.0000	91.0000

What-IF Prediction Choose Box

Choose the Prediction according to which System generates the next prediction, **SME** (Subject Matter Expert): decision solely based on users instinct and knowledge of business process about the process, **Prediction n** : decision solely based on the respective ranked confidence of the process

○ SME ○ Prediction 1 ○ Prediction 2 ● Prediction 3

Process Historical Behaviour

	Activity	Role	Time	Choice
0	Leucocytes	Role 4	-0.1194	Prediction 2
1	Leucocytes	Role 2	155.8796	Prediction 3

Predicted Time Duration of Predictions

	2
Expected	48.6670

1st Prediction

Activity

	Predicted	Confidence
2	ER Sepsis Triage	45.8405

Role

	Predicted	Confidence
2	Role 2	85.9531

Label

Regular

2nd Prediction

Activity

	Predicted	Confidence
2	ER Triage	28.3613

Role

	Predicted	Confidence
2	Role 4	11.2940

Label

Regular

3rd Prediction

Activity

	Predicted	Confidence
2	LacticAcid	5.5437

Role

	Predicted	Confidence
2	Role 5	2.6437

Label

Regular

Figure 5.12: What-If Mode UI

5.4.2.1 SME :

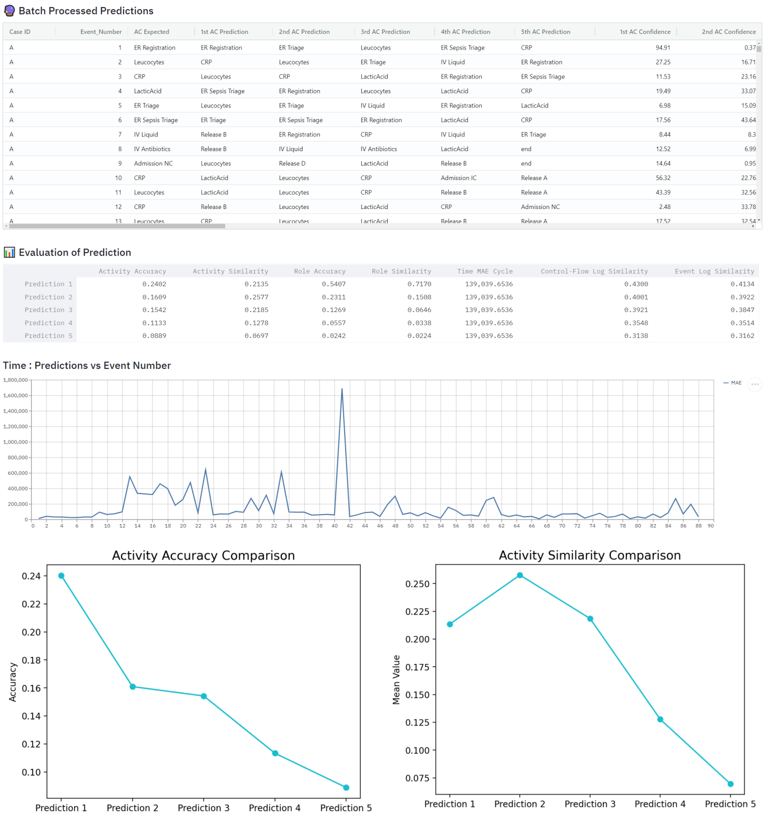
This option is exactly similar to the Base Mode (Subsection 5.4.1) when the pre-select prefix is zero. The mode differs because it lets the pre-selected prefixes to be incorporated. It is a log-level emulation of End of Trace — SME (Subsection 5.2.1.1) when the pre-select prefix is not zero, otherwise it replicates the Execution Mode (Section 5.1.1).

5.4.2.2 Prediction

In this option, the last event in the prefix is replaced by the prediction made in the previous event, i.e., for a trace of length n from the executed log, if l is the length of the prefix, then hd^{l-1} prefix inputs are taken from the test log and the l^{th} event in the prefix is the prediction of the previous event which is used to predict the next event. Thus, for the prefix $< e_1, e_2, ..., e_{l-1}, e_l >$, the prefix of first $l-1$ events $< e_1, e_2, ..., e_{l-1} >$ is extended with the prediction $e^{'}$ the model predicted using $< e_1, e_2, ..., e_{l-1} >$ resulting in the prefix $< e_1, e_2, ..., e_{l-1}, e^{'} >$. It is a log-level replication of the One-Step ahead Process Outcome (Subsection 5.1.1.1).

5.4.2.3 Generative

This option exactly imitates the End of Trace — Generative (Subsection 5.2.1.2) on a log level. The prefixes are solely from the predictions made so far. Since, the model is capable of generating complete traces of processes from the beginning with zero-prefix which is accomplished by continuous feedback in the form of newly generated events until it reaches the end of trace event also known as hallucination method.



🔮 Batch Processed Predictions

Case ID	Event_Number	AC Expected	1st AC Prediction	2nd AC Prediction	3rd AC Prediction	4th AC Prediction	5th AC Prediction	1st AC Confidence	2nd AC Confidence
A	1	ER Registration	ER Registration	ER Triage	Leucocytes	ER Sepsis Triage	CRP	94.91	0.37
A	2	Leucocytes	CRP	Leucocytes	ER Triage	IV Liquid	ER Registration	27.25	16.71
A	3	CRP	Leucocytes	CRP	LacticAcid	ER Registration	ER Sepsis Triage	11.53	23.16
A	4	LacticAcid	ER Sepsis Triage	ER Registration	Leucocytes	LacticAcid	CRP	19.49	33.07
A	5	ER Triage	Leucocytes	ER Triage	IV Liquid	ER Registration	LacticAcid	6.98	15.09
A	6	ER Sepsis Triage	ER Triage	ER Sepsis Triage	ER Registration	LacticAcid	CRP	17.56	43.64
A	7	IV Liquid	Release B	ER Registration	CRP	IV Liquid	ER Triage	8.44	8.3
A	8	IV Antibiotics	Release B	IV Liquid	IV Antibiotics	LacticAcid	end	12.52	6.99
A	9	Admission NC	Leucocytes	Release D	LacticAcid	Release B	end	14.64	0.95
A	10	CRP	LacticAcid	Leucocytes	CRP	Admission IC	Release A	56.32	22.76
A	11	Leucocytes	LacticAcid	Leucocytes	CRP	Release B	Release A	43.39	32.56
A	12	CRP	Release B	Leucocytes	LacticAcid	CRP	Admission NC	2.48	33.78
A	13	Leucocytes	CRP	Leucocytes	LacticAcid	Release B	Release A	17.52	32.54

📊 Evaluation of Prediction

	Activity Accuracy	Activity Similarity	Role Accuracy	Role Similarity	Time MAE Cycle	Control-Flow Log Similarity	Event Log Similarity
Prediction 1	0.2402	0.2135	0.5407	0.7170	139,039.6536	0.4300	0.4134
Prediction 2	0.1609	0.2577	0.2311	0.1508	139,039.6536	0.4001	0.3922
Prediction 3	0.1542	0.2185	0.1269	0.0646	139,039.6536	0.3921	0.3847
Prediction 4	0.1133	0.1278	0.0557	0.0338	139,039.6536	0.3548	0.3514
Prediction 5	0.0889	0.0697	0.0242	0.0224	139,039.6536	0.3138	0.3162

Time : Predictions vs Event Number

Figure 5.13: Batch Processing UI — Part I

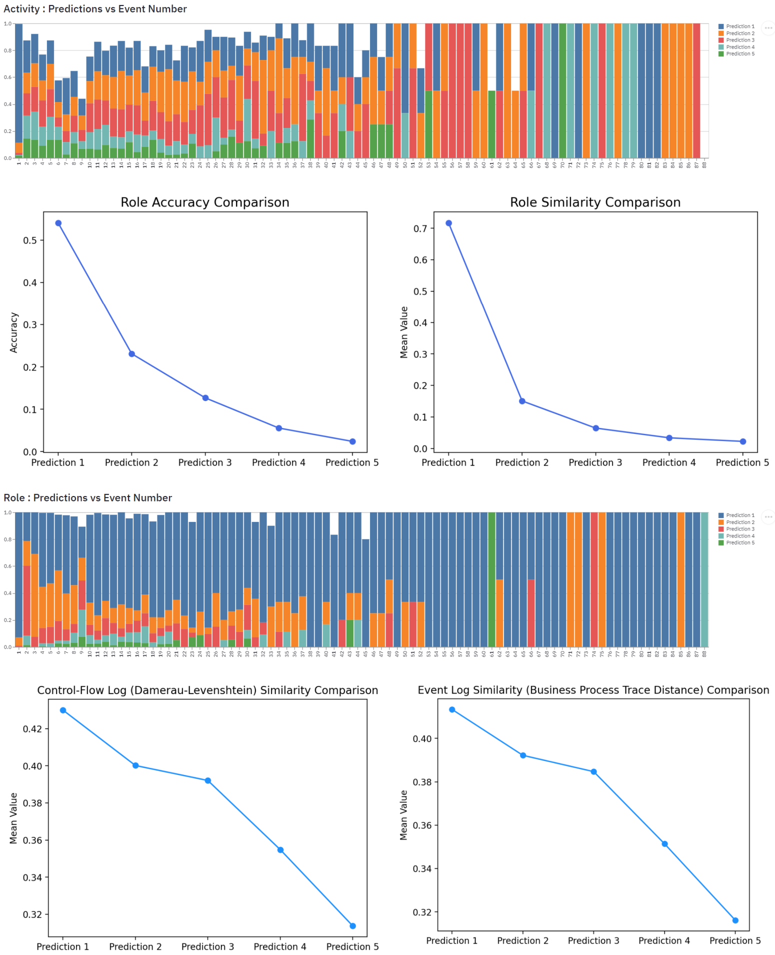


Figure 5.14: Batch Processing UI — Part II

5.4.3 Batch Processing User Interface

All the modes in the Batch Processing share the same user interface. Figure 5.13 and 5.14 combined is the developed UI of Batch processing. At the top it has the Batch processed prediction table which is capable of sorting and filtering. The next table tabulates the evaluation measure employed to measure the similarity on event and log level. Following that, it has the visualization to compare using different similarity measure for activity, role and time duration among the different predictive recommendations.

5.5 Summary

This chapter discussed the system frontend design of different modes of Single event processing and the Batch Processing. The One-Step Ahead prediction technique was introduced on the Execution mode to assist the user in making decisions at runtime. The Evaluation mode design discussed how the user can assess the model quality and the risks associated with it. Apart from this, our What-if mode design was also introduced, which let the user perform AI actions on a running trace and also simulate on the cases from the test log. The next chapter evaluates the multi-predictive recommendations presented in the Execution and Evaluation modes in the Batch Processing, utilizing the dashboard to investigate and quantify the multi-predictive recommendations.

Chapter 6

Dashboard Quantitative Evaluation of Predictive Techniques

In this chapter, we present an evaluation done using the dashboard Batch Processing, which evaluates the multi-prediction among different predictive recommendation techniques discussed in Chapter 5. All the evaluations are performed over the Sepsis Cases test event log in Pre-Selected Prefix Mode (5.4.2) across all options (Subsection 5.4.2.1, 5.4.2.2, 5.4.2.3), of which each represent different predictive recommendation techniques on a log level. We evaluate the test log using event level similarity for activity and roles, find the mean absolute error for time duration and two log level similarity measures. This chapter will address the following research question:

Research Question : How does the multi recommendations perform with each other?

The following sections comprise the chapter: — Section 6.1 introduces the log level similarity measure, Section 6.2 provide brief statistics on the Sepsis Cases test event log, then Section 6.3 provides the evaluation SME option of Pre-Selected Prefix Mode, this is followed by Generative option evaluation in Section 6.4, after that in Section 6.5 the Predictive option is evaluated. Finally, Section 6.6 compares the log level similarity among different predictive techniques evaluated using different options.

6.1 Evaluation Measures

The two evaluation techniques employed in Section 3.7.1: Damerau-Levenshtein (DL) distance and Mean Absolute Error (MAE) has been incorporated to measure the similarity between the pair of logs for Activity, Role, and Time with ground truth and the generated log. Both the evaluation is performed on the event level, i.e, similarity is calculated at each event number across the case and grouped together on event number. Then the mean is taken over the event number to calculate the activity and role similarity in terms of DL distance and MAE for time duration. The measurement of similarity on the log level is performed using Control-Flow Log Similarity (CFLS) [22] and Event Log Similarity (ELS) [31] measures. Both have also been incorporated in the evaluation study and discussed below.

`Control-Flow Log Similarity (CFLS)` is the measure of distance between two traces, one coming from the actual event log and the other generated from the predictive method employed. Using the inverse of the Damerau-Levenshtein (DL) distance similarity approach for activities (Section 3.7.1), each trace from the generated log is paired with the trace in the original event log such that it maximizes the sum of the trace similarities. This pairing is accomplished using the Hungarian algorithm[1] for determining ideal matches [36]. Subsequently, the CFLS is calculated as the average similarity of the best-matched traces.

`Event Log Similarity (ELS)` combines the control-flow similarity measure and the time-perspective measure (MAE) as defined in Camargo et al. [31]. The approach of measuring the ELS is similar to CFLS, except instead of using DL distance to measure similarity between traces, it uses Business Process Trace Distance (BPTD), which considers the activity and the timestamp. Similar to the CLFS distance measure, BPTD takes the concurrency into account for activities, and in the case of the activities label match, the penalty is proportional to the difference in the timestamp. This penalty is standardised between 0 and 1. While the BPTD enables the comparison of two traces, when applied to an event log, it incorporates the Hungarian algorithm [36] matches with the traces that minimises the sum of the BPTDs of the paired traces. This similarity measure is referred to as Event Log Similarity (ELS).

6.2 Understanding Test Event Log

The Sepsis test event log consists of 229 cases, of which there are 3812 events. The statistics of the test log is tabulated in Table 6.1. The detailed information about the event log has been discussed in Section 3.2.

It is important to note that the cases >24 events constitute less than 10% of the number of cases. This hinders the generalization of the long events, although splitting the log into small and long traces would add too much variability to compare, so we restricted it to just using as it is.

[1] Hungarian Algorithm: `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html`.

Table 6.1: Statistics of Sepsis Test Event Log

#cases	#activities	# events	case length			case duration (days)		
			min	max	avg.	min	max	avg.
229	16	3812	8	88	16.64	0.12	241.14	22.6

6.3 Evaluation — Using SME Option

In this section, we will evaluate the Execution mode on a log level using SME option in Pre-Selected Prefix Mode. It's Next Action recommendation capabilities are quantified over Sepsis Cases test event log. The design was discussed in Section 5.1.1 and 5.2.1.1 on a case level. Here, subsequent actions are recommended based on the feedback from the ongoing case (event test log) in the form of the prefix to the model.

Table 6.2 reports the similarity measure using maxima (arg max) where it compares different predictive recommendations based on prefix size. Prediction 1 to Prediction 5 is ordered in descending order of their probability of occurrence (confidence). Prediction 1 performs best with zero pre-selected prefixes overall, whereas for Prediction 2 and 3, similarity increases as the prefix size increases. Prediction 4 and 5 showed quite random behaviour, although the similarity is very low at event level similarity measurement. MAE showed consistent behaviour across different pre-select prefix predictions, where 2^{nd} prefix performed best across the predictions. However, the MAE difference is relatively small across the prefixes.

It is apparent that for the event log tested, an increase in the pre-selected prefix does not help in the case of the highest probability predicted category (Prediction 1). Next in the Table 6.3 we see the comparison among the different prediction, it has been averaged out among the pre-select prefix length of Table 6.2 for each prediction.

Prediction 1 (Highest) outperforms all the predictions in the role similarity measure. However, the event level similarity measure of activity performs way better for Prediction 2. Prediction 1 and 3 performed quite similar for the activity similarity. Prediction 4 and 5 have pretty equivalent event-level similarities but are not significant enough comparatively. CFLS and ELS measure is best for Prediction 1. Here, more similarity means that the user will be recommended similar actions on the dashboard as it was in the event log, but it will also lead to have few different options for that predictive measure that they could think are right based on their business acumen.

Figure 6.1 shows the DL similarity of activity at different event numbers for

Table 6.2: SME Predictive Recommendation for different size prefix

Pred_Number	Prefix	Activity Similarity (DL)	Role Similarity (DL)	Time MAE Cycle (Days)	Control-Flow Log Similarity	Event Log Similarity
Prediction 1	0	**0.2636**	**0.8247**	1.6792	**0.6367**	**0.6123**
	2	0.2484	0.8220	**1.6576**	0.5922	0.5647
	4	0.2405	0.8223	1.6829	0.5741	0.5404
	6	0.2367	0.8213	1.7143	0.5700	0.5310
Prediction 2	0	0.3623	0.1204	1.6792	0.4398	0.4273
	2	0.3700	**0.1224**	**1.6576**	0.4929	0.4792
	4	0.3759	0.1210	1.6829	**0.5273**	**0.5125**
	6	**0.3780**	0.1212	1.7143	0.5136	0.4926
Prediction 3	0	0.2276	0.0408	1.6792	0.3740	0.3597
	2	0.2326	0.0412	**1.6576**	0.4198	0.4019
	4	0.2352	0.0420	1.6829	**0.4420**	**0.4191**
	6	**0.2361**	**0.0425**	1.7143	0.4260	0.3971
Prediction 4	0	0.0513	0.0079	1.6792	0.3241	0.3194
	2	**0.0524**	0.0081	**1.6576**	**0.3508**	**0.3453**
	4	0.0503	0.0083	1.6829	0.3395	0.3246
	6	0.0496	**0.0084**	1.7143	0.3240	0.3021
Prediction 5	0	0.0466	0.0004	1.6792	**0.3396**	**0.3405**
	2	0.0475	0.0004	**1.6576**	0.3112	0.3086
	4	0.0480	**0.0004**	1.6829	0.3218	0.3186
	6	**0.0487**	0.0004	1.7143	0.3121	0.2923

Table 6.3: SME Predictive Recommendation of different predictions

Pred Number	Activity Similarity (DL)	Role Similarity (DL)	Control-Flow Log Similarity	Event Log Similarity
Prediction 1	0.2473	**0.8226**	**0.5933**	**0.5621**
Prediction 2	**0.3716**	0.1212	0.4934	0.4779
Prediction 3	0.2329	0.0416	0.4154	0.3945
Prediction 4	0.0509	0.0082	0.3346	0.3228
Prediction 5	0.0477	0.0004	0.3212	0.3150

each prediction. It is quite clear that Prediction 1 performed consistently best for the first three events, following that Prediction 2 performed similar to Prediction 1 and beyond 45^{th} event Prediction 1 similarity measure has almost vanished. As mentioned earlier in Section 6.2, >24 events constitute less than 10% of number of cases in the test log. Thus, it is quite hard to judge the similarity for later events. Nevertheless, Prediction 2 and 3 is quite prevalent for long traces.

Overall, Prediction 1 performs best concerning similar predictions as it was in test event log, but regarding predictive recommendations, it needs other predictions to assist the user to recommend the other possible outcomes which are likely possible.

Figure 6.2 shows the comparison of maxima predictive recommendations with the random. To compute random, each set of pre-select prefixes was run five times, and then the mean was taken over them. It is then averaged out over all the prefixes for respective predictions for each measure before the comparison. For the similarity measure of activity and role, maxima performed better (Figure 6.2a, Figure 6.2b). However, the random performed better in the case of the highest probability prediction of activity (Prediction 1) as depicted in Figure 6.2a. CFLS

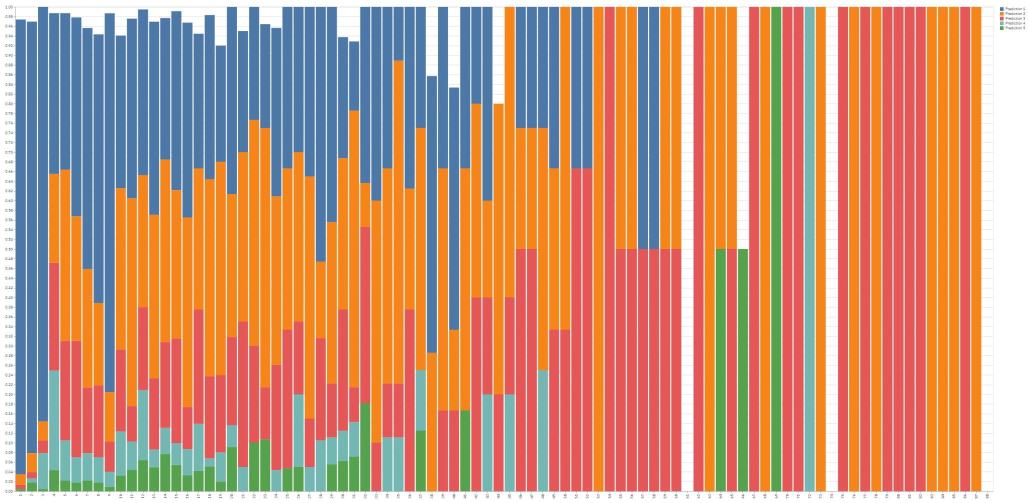


Figure 6.1: SME Activity Similarity over different Event Number

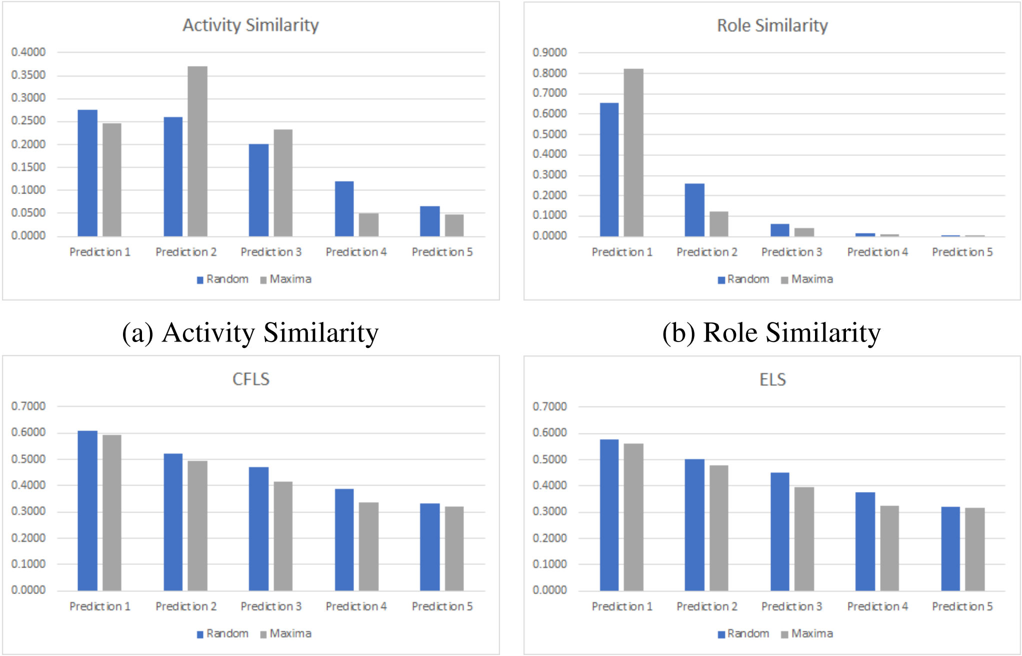
and ELS performed better by a slight margin in the case of random, which implies maxima would lead to have different set of recommendation to choose from over the random, whereas random might be emulating more close to the event log behaviour. Although, difference between log level measurement is not quite high so in case of sepsis event log the difference might not be prevalent in Single Event Processing.

6.4 Evaluation — Generative Option

This section will assess the generative recommendation behaviour of the model, which is capable of predicting the next action on its own. The generated predictions are fed as the feedback (prefix) to the model to generate a complete trace of the process. The evaluation uses Generative option in Pre-Selected Prefix Mode and the predictive design has been discussed in End of Trace — Generative (Section 5.2.1.2) and One-Step ahead Generative Outcome (Section 5.1.1.2).

Table 6.4 lists the similarity measures of different prefix lengths of each of the predictions using maxima (arg max). As explained in Section 5.2.1.2, unlike predictive recommendation evaluated in SME option where all the feedback (prefix) are coming from one source (test event log), in generative mode, each feedback is independent of each other because each predicted recommendation act as the independent prefix which is fed to the predictive model to recommend the next action.

In Prediction 1, zero pre-select prefix performed better overall concerning similarity measure. Prediction 2 performed better as the pre-select prefix length



(a) Activity Similarity

(b) Role Similarity

(c) CFLS

(d) ELS

Figure 6.2: Random & Maxima SME Predictive Recommendation Similarity

is increased, although it showed a reduction in the similarity for the pre-select prefix of length 6. Prediction 3 and 4 showed similar behaviour on event-level similarity measurement, i.e., with the increase in prefix length activity and role similarity increases, whereas the zero-pre-select prefix is measured high for event log similarity measurement. Prediction 5 shows the sporadic behaviour among all the pre-select prefixes, but all the similarity measure values are pretty similar. MAE measure performed consistently best across the prediction for the pre-select prefix of length two, although all the MAE is quite similar to each other, and there is not much of a difference among them.

In Table 6.5, each prediction's prefix length has been averaged over respective predictions in order to make a comparison among them. Prediction 1 performed best among all the similarity measures. However, the other similarity measures are also quite similar to each other. This might not be quite helpful when recommending actions on the dashboard because there is more chance of recommending the same actions across different predictive recommendations, which limits the options provided to the user, on the other side it builds confidence when it is being predicted as a probable outcome of the chosen action.

All in all, Prediction 1 performs the best concerning generative prediction, but in terms of recommending other probable outcomes, it may not require other

Table 6.4: Generative Predictive Recommendation for different size prefix

Pred_Number	Prefix	Activity Similarity (DL)	Role Similarity (DL)	Time MAE Cycle (Days)	Control-Flow Log Similarity	Event Log Similarity
Prediction 1	0	**0.2035**	**0.5343**	1.8495	**0.6340**	**0.6128**
	2	0.1891	0.5278	**1.8322**	0.5884	0.5639
	4	0.1818	0.5253	1.8616	0.5773	0.5435
	6	0.1791	0.5196	1.8996	0.5775	0.5348
Prediction 2	0	0.1554	0.4946	1.8510	0.5539	0.5251
	2	0.1760	0.5110	**1.8360**	0.5043	0.4742
	4	**0.1817**	**0.5153**	1.8619	**0.5733**	**0.5397**
	6	0.1717	0.5105	1.8977	0.5424	0.5108
Prediction 3	0	0.1657	0.5031	1.8475	**0.5562**	**0.5276**
	2	0.1749	0.5082	**1.8353**	0.5070	0.4773
	4	**0.1772**	**0.5111**	1.8653	0.5249	0.4960
	6	0.1663	0.5037	1.9015	0.5123	0.4897
Prediction 4	0	0.1600	0.5005	1.8525	**0.5618**	**0.5368**
	2	0.1767	0.5090	**1.8373**	0.5126	0.4839
	4	**0.1772**	**0.5111**	1.8653	0.5249	0.4960
	6	0.1663	0.5037	1.9015	0.5123	0.4897
Prediction 5	0	0.1711	0.5066	1.8479	0.5351	**0.5110**
	2	**0.1796**	0.5169	**1.8346**	0.5321	0.5108
	4	0.1754	**0.5114**	1.8584	0.5269	0.4984
	6	0.1724	0.5076	1.8991	**0.5361**	0.5081

Table 6.5: Generative Predictive Recommendation of different predictions

Pred Number	Activity Similarity (DL)	Role Similarity (DL)	Control-Flow Log Similarity	Event Log Similarity
Prediction 1	**0.1884**	**0.5267**	**0.5943**	**0.5637**
Prediction 2	0.1712	0.5079	0.5435	0.5124
Prediction 3	0.1710	0.5065	0.5251	0.4976
Prediction 4	0.1701	0.5061	0.5279	0.5016
Prediction 5	0.1746	0.5106	0.5325	0.5325

predictive recommendations because they are quite similar quantitatively, which reduces the variability in the predictive recommendations.

Figure 6.3 shows the similarity distribution of activity of different predictions over the respective event number for the event log with the zero-pre-select prefix. Prediction 1 performed better initially, but as the event progressed, other predictions became equally similar. However, beyond the 64^{th} event number, none of the predictive recommendations was able to predict a similar activity as it is in the event log, also, for the events between 44^{th} to 63^{rd} there are quite hit-and-miss. However, as discussed earlier, > 24 events account for less than 10% of the total number of instances in the test log. As a result, determining the generalization of these occurrences is rather difficult to state.

Figure 6.4 gives the similarity comparison of random and maxima method of generative predictive recommendation for different predictions. The computation of random similarity measure has been done in the same way as it was discussed in Section 6.5. Maxima performed best in all the similarity comparisons, and activity similarity difference is relatively less in comparison to the role similarity measure across the different predictions. CFLS and ELS measurement also do

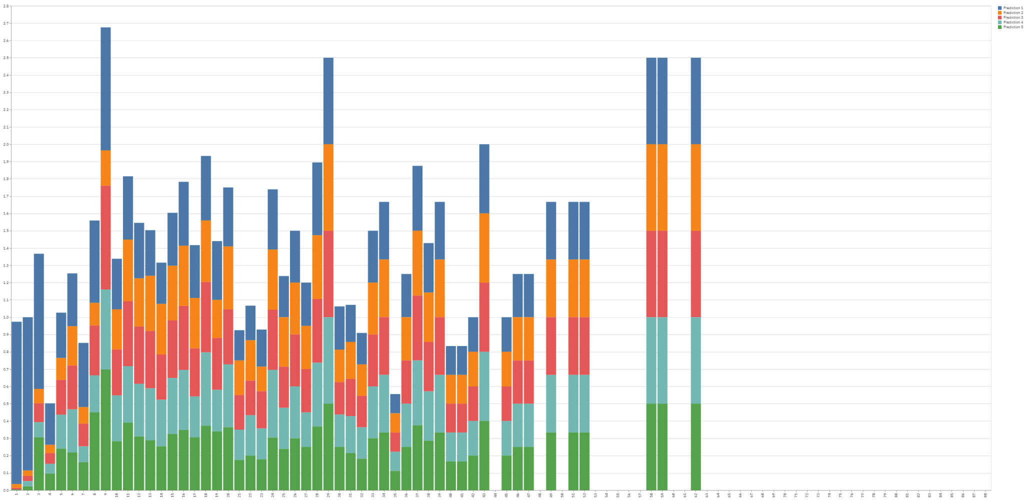


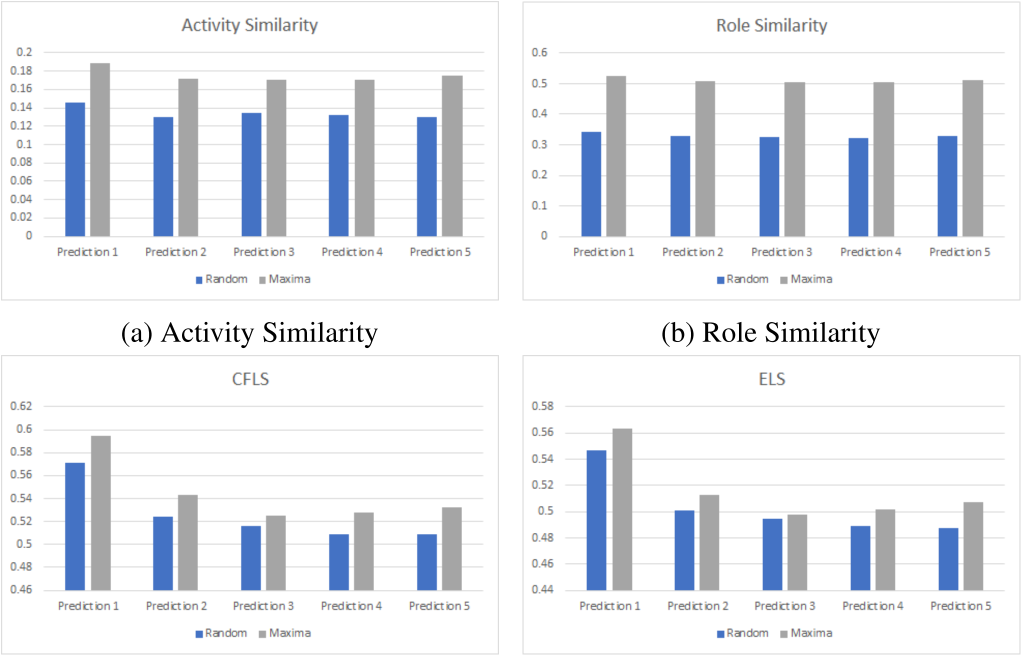
Figure 6.3: Generative Activity Similarity over different Event Number

not differ by considerable value between the random and maxima across the predictions. Thus, a generative predictive recommendation based on maxima predictive recommendation is more similar in emulating the event log than the random recommendation.

6.5 Evaluation — Prediction Option

The following section will evaluate the predictive recommendation discussed in One-Step ahead Process Outcome (Section 5.1.1.1) using Predictive option in Pre-Selected Prefix Mode. The main difference here is that it replaces the last event of the prefix coming from the event log with the previous step predicted event. It relies on the idea that the last element is the most influential factor in determining what would be predicted next in the series prediction. This mode also leads into the similar situation as for the generative predictive recommendation, where all the prefixes act independent of each other.

A comparison of predictive predictions based on varied prefix lengths are reported in Table 6.6 using maxima (arg max) similarity measure. The zero-pre-select prefix outperformed all other prefixes in terms of similarity in Prediction 1. Prediction 2 and 3 performed better as the number of pre-select prefixes was increased, although it indicated a decline in similarity for the sixth pre-select prefix. Role similarity performed better for the pre-select prefix of length two in Prediction 2 and 3. Prediction 4 and 5 showed relatively similar behaviour. Activity and role similarity measure highest for pre-select prefix of length two for the event level measurement, while CFLS and ELS similarity measure are best for



(a) Activity Similarity

(b) Role Similarity

(c) CFLS

(d) ELS

Figure 6.4: Random & Maxima Generative Predictive Recommendation Similarity

the zero-pre-select prefix.

The difference in similarity between all of the pre-select prefixes for the respective predictions is relatively small. Despite the fact that all MAE are reasonably equivalent and have the minimal difference, the MAE measure continuously performed the best throughout either for the pre-select prefix of length two and three.

In the Table 6.7 compares among different Predictions regarding similarity measure. Prediction 1 similarity measure is best among all similarity parameters. However, the difference among the measure is quite low, which indicates that the recommended actions on the dashboard will not differ much, which would limit the user to select different recommendations, although, in the case of one step ahead prediction, this would lead to more confidence with the selected predictive action.

As seen in Figure 6.5, the similarity distributions of activity of multi-predictions during the course of each event of the event log with the zero-pre-select-prefix. Although Prediction 1 fared better initially, other forecasts were equally relevant as the event progressed. At 7^{th} event, all the predictions seem to perform equally well. However, after the 59^{th} event onwards, the similarity measure shows very erratic behaviour. Nevertheless, as discussed in previous sections, cases > 24 events account for fewer than 10% in the test log. As a consequence, generalizing similarity measure over long events is tricky.

Table 6.6: Process Predictive Recommendation for different size prefix

Pred_Number	Prefix	Activity Similarity (DL)	Role Similarity (DL)	Time MAE Cycle (Days)	Control-Flow Log Similarity	Event Log Similarity
Prediction 1	0	**0.2211**	**0.4880**	2.3552	**0.6366**	**0.6105**
	2	0.2063	0.4792	2.3497	0.5903	0.5603
	4	0.1978	0.4712	**2.3911**	0.5733	0.5374
	6	0.1966	0.4669	2.4313	0.5788	0.5364
Prediction 2	0	0.1994	0.4663	2.3617	0.5311	0.5084
	2	0.1959	**0.4694**	**2.3517**	0.5233	0.4944
	4	**0.2004**	0.4677	2.3827	**0.5706**	**0.5360**
	6	0.1929	0.4575	2.4279	0.5416	0.5089
Prediction 3	0	0.1926	0.4597	2.3552	0.5391	0.5133
	2	0.1961	**0.4683**	**2.3502**	0.5275	0.4988
	4	**0.1987**	0.4668	2.3805	**0.5666**	**0.5337**
	6	0.1902	0.4539	2.4349	0.5326	0.5027
Prediction 4	0	0.1934	0.4610	2.3586	**0.5381**	**0.5124**
	2	**0.1968**	**0.4686**	**2.3508**	0.5348	0.5064
	4	0.1957	0.4637	2.3848	0.5303	0.4976
	6	0.1881	0.4524	2.4317	0.5184	0.4937
Prediction 5	0	0.1926	0.4602	2.3533	**0.5465**	**0.5281**
	2	**0.1958**	**0.4681**	**2.3513**	0.5341	0.5051
	4	0.1960	0.4656	2.3823	0.5264	0.4951
	6	0.1900	0.4548	2.4302	0.5437	0.5134

Table 6.7: Process Predictive Recommendation of different predictions

Pred Number	Activity Similarity (DL)	Role Similarity (DL)	Control-Flow Log Similarity	Event Log Similarity
Prediction 1	**0.2054**	**0.4763**	**0.5947**	**0.5612**
Prediction 2	0.1972	0.4652	0.5417	0.5119
Prediction 3	0.1944	0.4622	0.5414	0.5121
Prediction 4	0.1935	0.4614	0.5304	0.5025
Prediction 5	0.1936	0.4622	0.5377	0.5104

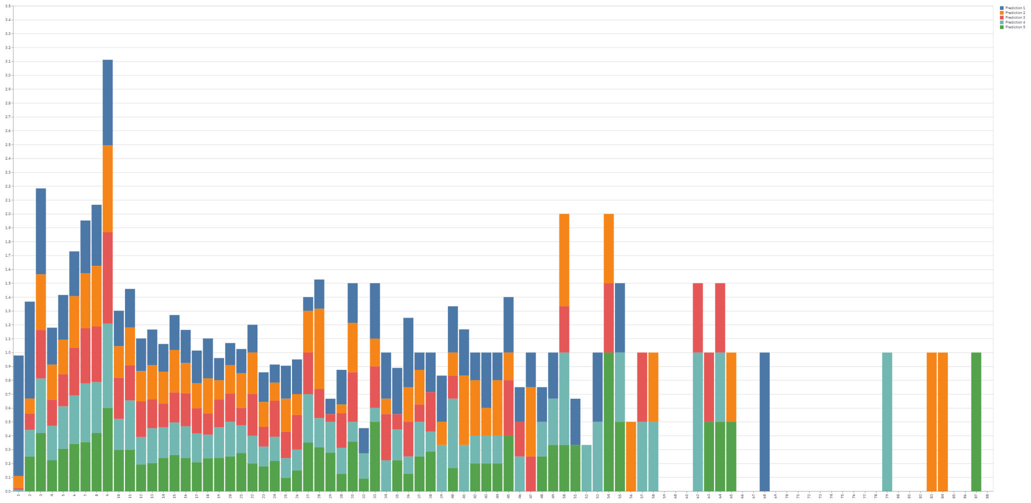
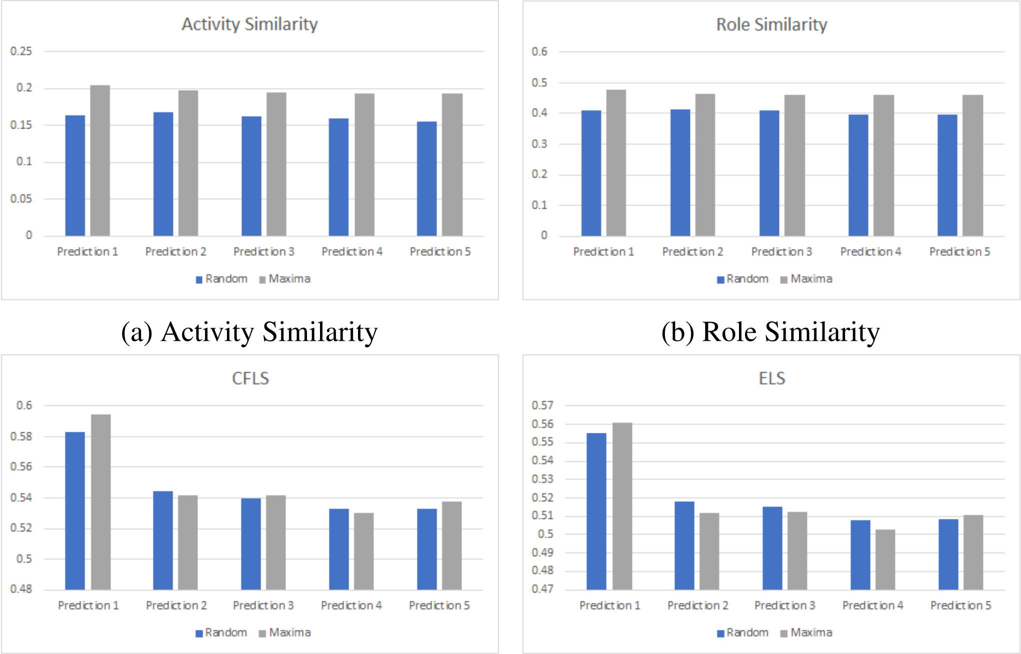


Figure 6.5: Process Activity Similarity over different Event Number

The comparison of maximum predictive recommendations to random is shown in Figure 6.6 for the process predictive recommendation. As discussed in Section 6.5, the random similarity measure was computed in the same manner. In the event-based similarity measure, the maxima for activity and role performed best, as shown in Figure 6.6a and 6.6b consistently by the same margin. However, in the CFLS (Figure 6.6c) and ELS (Figure 6.6d) measure maxima Prediction 1 performed better, but the random measured better for Prediction 2 and 3 for the CFLS and Prediction 5 for ELS measure. So, maxima's predictive recommendation is not always advisable, but random recommendation does not guarantee a confident better recommendation either.



(a) Activity Similarity

(b) Role Similarity

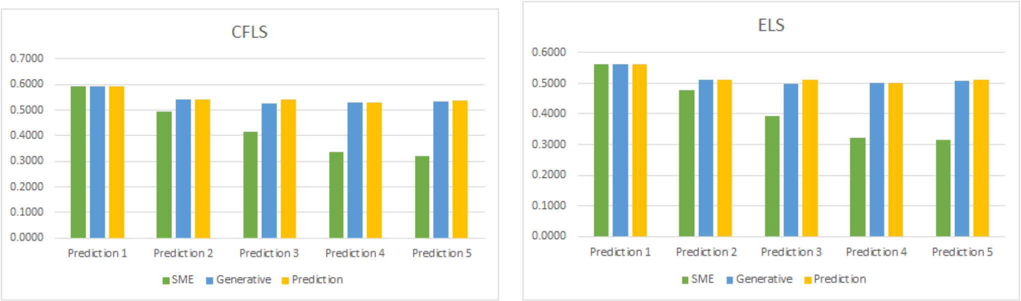
(c) CFLS

(d) ELS

Figure 6.6: Random & Maxima Process Predictive Recommendation Similarity

6.6 Discussion

Figure 6.7 depicts the comparison among different predictive recommendations in maxima for the SME, Generative and Predictive options. Prediction 1 is quite similar across the options. Although, for SME option similarity measure is the least among other predictions, whereas Generative and Predictive option performed quite similar. The reason behind Generative and Predictive to have quite similar behaviour because both work on the same principle of having independent prefixes for each recommendation by design.



(a) CFLS

(b) ELS

Figure 6.7: Log level similarity comparison between SME, Generative and Predictive

This comparison helps us to recognize that, in the case of just recommending the most probable outcome by LSTM model, all three techniques been evaluated by SME, Generative and Predictive options will turn out same. Although when we look at multi-predictive recommendation perspective, it has to recommend different predictive actions. Predictive techniques used in Generative and Predictive option might lead to recommend same actions across in the multi predictive scenario. Whereas, the predictive technique used in SME option will lead to recommend different recommendations which is what we are aiming for while recommending the actions, as our assumption is that the user will apply the business acumen to select the action among the recommendations provided on the dashboard. That doesn't mean the predictive techniques of Generative and Predictive options can't be of any use. As we have discussed earlier, predictive techniques of Generative and Predictive options aren't used for recommending any actions, rather it is used to assist user to see what possibilities are because of the actions being selected and understand what model has seen before while training. Thus, our hypothesis is :

Hypothesis 1. Predictive techniques of Generative and Predictive options, i.e, One-Step ahead Process Outcome, One-Step ahead Generative Outcome and End of Trace — Generative, might recommend same prediction in the multi-predictive recommendation behaviour for ≥ 2 predictions for certain events.

6.7 Summary

This chapter discussed the evaluation of different predictive techniques used and the evaluation measurement. The evaluation consists of the measurement of similarity across the event level as well as on the log level using CFLS and ELS. During the evaluation, we saw how the similarity measure could limit the multi-predictive recommendations but also could be used to build the confidence in the user to take action if used in the form to see the foreseeable future for the chosen recommendation. In the next chapter, we will demonstrate the Single Event Processing modes and verify our hypothesis.

Chapter 7

Dashboard at Runtime: Single Event Processing Demonstration

So far, we have addressed the dashboard architecture, design of the predictive techniques of the system, and dashboard based quantitative evaluation on the Sepsis case test event log. In this chapter, we will focus on how the dashboard will assist the business user to make decisions at runtime, with the assumption that *they have an understanding of the business and the consecutive actions which are supposed to be taken*. We will be using different modes of Single Event Processing, and the following research question will be addressed in this chapter:

Research Question : How to use, translate, and explain the confidence on the recommendations provided by the model with the available information from the dashboard?

This chapter is divided into the various demonstration of different modes of Single Event Processing and has the following sections — Section 7.1 discusses the Execution mode demonstration and how the dashboard can assist the user for an ongoing case, Section 7.2 deals with building trust and confidence in the model from the user's perspective using the Evaluation mode, and finally, in Section 7.3, we go through the demonstration of two cases of execution, trying to simulate each other's execution sequence in the What-If mode.

7.1 Demonstration — Runtime Action

This section explores the convenience of the dashboard for a user to make decisions, using the Sepsis test event log at runtime, with the Execution Mode. Table 7.1 tabulates the next activity of the case-id "WFA" from sepsis. In this demonstration, we emulate how a business user might interpret the different predictive recommendations that are offered by the dashboard. Using a concrete history trace for each prefix, we will evaluate the predictions generated and how well they correlate with different predictive recommendations shown on the dashboard or with the log process behaviour depicted in Figure 3.2. So,

- If the next event that the user selects is also shown on the dashboard's predictive recommendation, we can conclude that the dashboard might have been beneficial to the user in this situation.

- If the next event that the user selects is not shown on the dashboard, we make a note of this as well.

One-step ahead predictions are meant to establish confidence and provide future insights to the user with all the possibilities as per the different predictive recommendations available on the dashboard. It is not the recommendation to act upon. All the activities are denoted by their corresponding activity index values to accommodate them in the report, along with their activity legends at the bottom.

Objective: Execution Mode demonstration is regarded successful if, when replying for each event, there is a recommendation that corresponds to the test log's historical actions. However, if the recommendation misses at any event, the demonstration will be considered unsuccessful. The reasoning behind this categorization into successful and unsuccessful for this demonstration is to indicate if the dashboard was able to provide basic support of guidance during the process execution.

Table 7.1 tabulates the top three recommendations for the selected case-id : "WFA". Each predictive recommendation (p_r) is provided with its confidence (c), and one step ahead prediction (p). The observations are described based on the hypothesis that the business user is aware of the most evident flow of events (Fig 3.2), i.e., the process based on which actions have to be taken.

Figure 7.1 shows how the dashboard presents the recommendations for event #1 (Table 7.1). The Model is executed after appending ER Registration (4) in the prefix. Thus, it is the Process History Behaviour which predicted ER Triage (6) with a confidence of 90.8%, CRP (3) with 2.9%, and Leucocytes (10) with 1.8%.

Figure 7.2 shows the one-step ahead prediction of event #1 (Table 7.1). The left side shows the One-Step ahead Process Outcome as ER Sepsis Triage (5), Leucocytes (10), and ER Sepsis Triage (5) for 1st, 2nd and 3rd predictions, respectively, and the right side shows the One-Step ahead Generative Outcome as ER Sepsis Triage (5), ER Registration (4), and ER Registration (10) for 1st, 2nd and 3rd predictions, respectively.

`#0 event`: As mentioned in Section 5.1.1, the first recommendation starts with zero-prefix, and it recommended ER Registration (4) with a confidence of

Table 7.1: Execution predictive output for the Case-id WFA

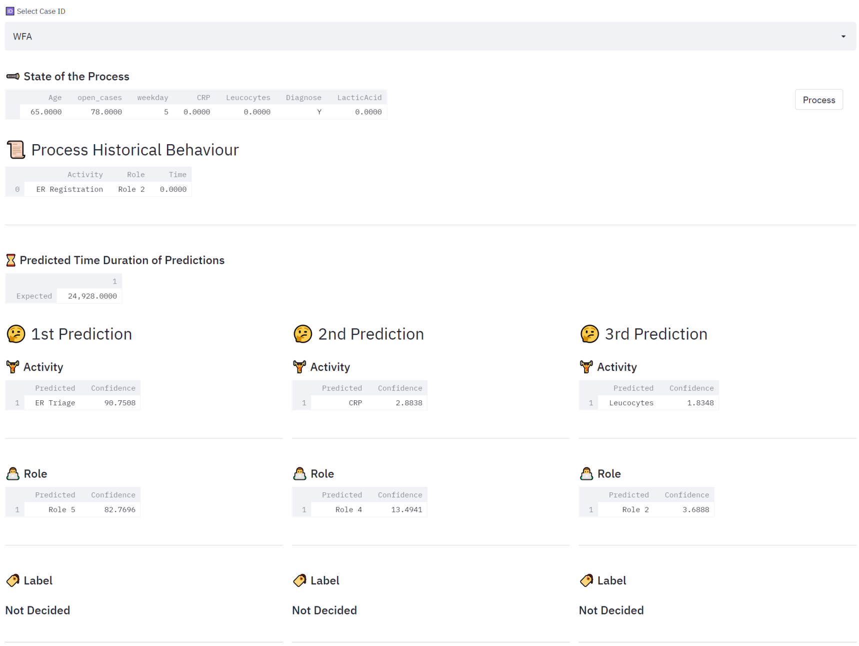
#event	Prefixes	Predicted Recommendation						One-Step ahead Process Outcome			One-Step ahead Generative Outcome		
		1st		2nd		3rd		1st	2nd	3rd	1st	2nd	3rd
		p_r	c	p_r	c	p_r	c	p	p	p	p	p	p
0	0	**4**	**97.2**	10	0.7	3	0.6	**6**	3	4	**6**	3	4
1	4	**6**	**90.8**	3	2.9	10	1.8	**5**	10	**5**	**5**	4	4
2	6	**5**	**91.2**	10	2.7	3	2.2	9	9	9	9	10	5
3	5	9	42.3	3	20.9	**8**	**17.5**	10	3	3	10	6	9
4	8	3	28.6	9	27.6	**7**	**21.4**	**10**	**10**	**10**	**10**	6	9
5	7	9	33.0	3	30.0	**10**	**23.3**	10	**9**	**9**	8	5	**9**
6	10	10	38.6	**9**	**30.4**	3	29.9	10	7	10	7	8	7
7	9	10	31.6	7	22.9	**3**	**16.0**	**2**	**2**	**2**	**2**	**2**	**2**
8	3	**2**	**88.0**	8	3.0	3	2.9	10	2	**3**	10	10	10
9	2	10	38.8	**3**	**38.7**	2	11.8	**10**	**10**	**10**	**10**	**10**	**10**
10	3	**10**	**49.2**	3	34.5	11	10.9	**11**	15	10	**11**	10	10
11	10	**11**	**38.7**	10	30.5	3	23.6	10	**15**	10	**11**	17	10
12	11	**15**	**63.2**	17	35.7	11	0.3	-	-	-	-	-	-
13	15	-	-	-	-	-	-	-	-	-	-	-	-

1st, 2nd, 3rd : top three respective recommendations and predictions.
p_r: recommendation ; c: confidence (measured in %) ; p: prediction
Activity Index — 1: Admission IC, 2: Admission NC, 3: CRP, 4: ER Registration, 5: ER Sepsis Triage, 6: ER Triage, 7: IV Antibiotics, 8: IV Liquid, 9: LacticAcid, 10: Leucocytes, 11: Release A, 12: Release B, 13: Release C, 14: Release D, 15: Return ER, 16: other, 0: start, 17: end
Bold Value Means :

- Predicted Recommendation : The recommended action was selected as prefix for next event. e.g., 1st recommendation of #0 event is the prefix for #1 event.
- One Step ahead : The selected prediction is chosen as an action in one step ahead. e.g., 1st prediction of both process and generative in #0 event is chosen as an action in #1 event and act as prefix to #2 event.

97.2%, Leucocytes (10) with 0.7%, and CRP (3) with 0.6% out of which, ER Registration had the highest confidence of 97.2%. Looking at the one-step-ahead outcomes, both types predicted ER Triage (6) as the next ideal action after the most confident recommendation. This indicates that the model has seen ER Triage as the only action occurring after ER Registration most of the time, which also fits in the main Sepsis Cases event log process behaviour, as shown in Figure 3.2. The user chose ER Registration (4) as the next action for this event, which is also the first recommendation. The choice was appended to the prefix of length zero.

`#1 event`: After having executed ER Registration (4), the dashboard shows the next most probable predictive recommendation as ER Triage (6) with a confi-



Select Case ID

WFA

State of the Process

	Age	open_cases	weekday	CRP	Leucocytes	Diagnose	LacticAcid
	65.0000	78.0000	5	0.0000	0.0000	Y	0.0000

Process Historical Behaviour

	Activity	Role	Time
0	ER Registration	Role 2	0.0000

Predicted Time Duration of Predictions

	1
Expected	24,928.0000

1st Prediction

Activity

	Predicted	Confidence
1	ER Triage	90.7508

Role

	Predicted	Confidence
1	Role 5	82.7696

Label

Not Decided

2nd Prediction

Activity

	Predicted	Confidence
1	CRP	2.8838

Role

	Predicted	Confidence
1	Role 4	13.4941

Label

Not Decided

3rd Prediction

Activity

	Predicted	Confidence
1	Leucocytes	1.8348

Role

	Predicted	Confidence
1	Role 2	3.6888

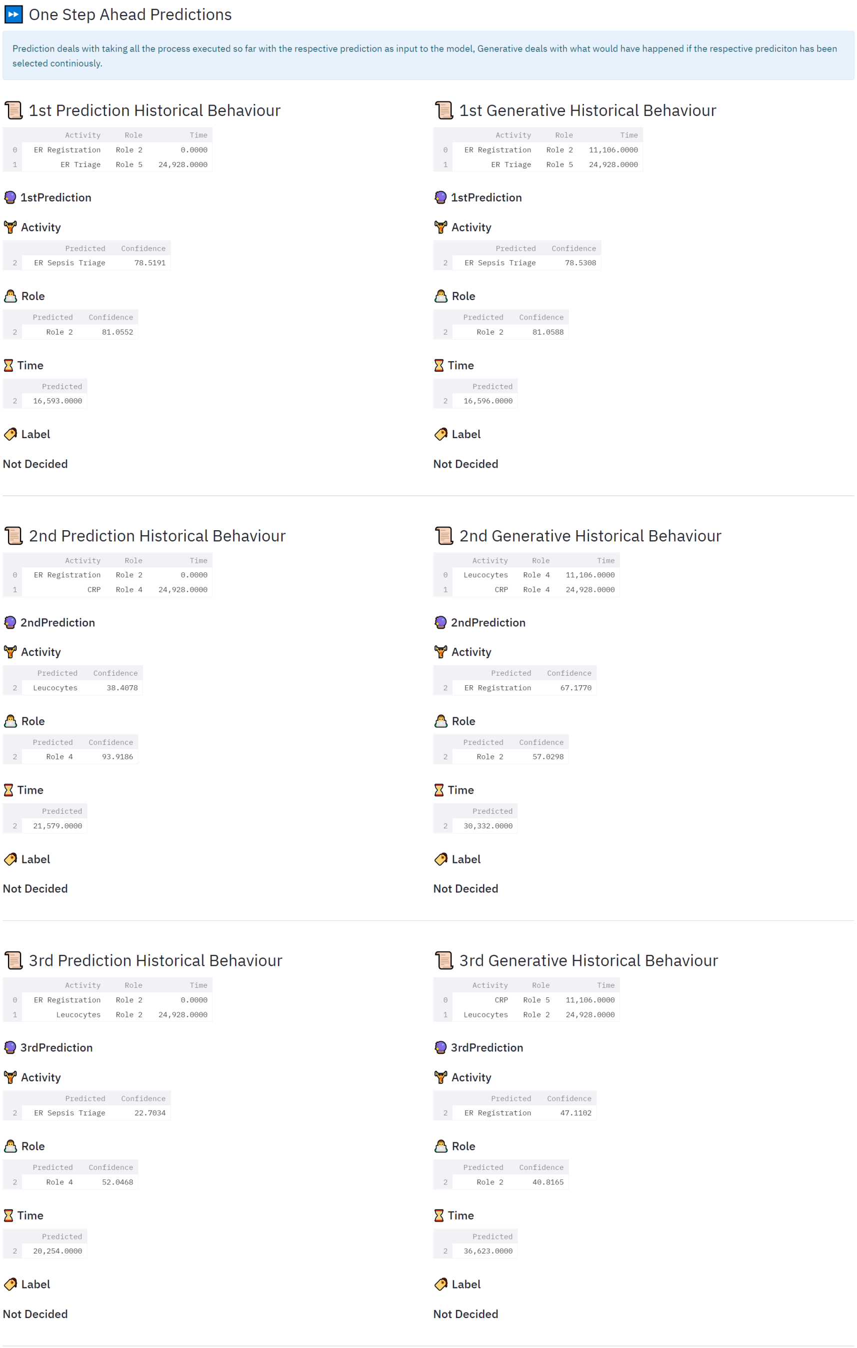
Label

Not Decided

Figure 7.1: Recommendation on the dashboard for #1 Event

dence of 90.8%, CRP (3) with 2.9%, and Leucocytes (10) with 1.8%. ER Triage having the highest confidence, can be selected as the next action, which was also as per the one-step-ahead prediction of event #0. ER Sepsis Triage (5) was predicted as one step ahead as the most reoccurring recommendation, which is per the process execution (Fig 3.2). Also, our hypothesis 1 turned out to be true in the case of One-step ahead process outcome, as it predicted ER Sepsis Triage (5) for the 1st and 3rd predictions and One-step ahead Generative outcome predicted ER Registration (4) for the 2nd and 3rd predictions. The user chooses ER Triage (6) as the action for this event which is as per the process flow (Fig 3.2), and the first recommendation. Thus, the prefix for the next event would be <ER Registration (4), ER Triage (6)>.

`#2 event`: After having executed ER Triage (6), the model recommended ER Sepsis Triage (5) with a confidence of 92.7%, Leucocytes (10) with 2.7%, and CRP (3) with 2.2%. ER Sepsis Triage had the highest confidence and can be picked as an action. None of the one-step ahead predictions were per the process execution flow (Fig 3.2), but our hypothesis 1 turned out to be true in the case of One-step ahead process outcome where it recommended the same prediction, LacticAcid (9), across three of its predictions. This indicates that irrespective of the action taken, there is a high probability of LacticAcid being recommended as the most confident action when predictive recommendations of this event are appended to the prefix



One Step Ahead Predictions

Prediction deals with taking all the process executed so far with the respective prediction as input to the model, Generative deals with what would have happened if the respective prediciton has been selected continiously.

1st Prediction Historical Behaviour

	Activity	Role	Time
0	ER Registration	Role 2	0.0000
1	ER Triage	Role 5	24,928.0000

1stPrediction

Activity

	Predicted	Confidence
2	ER Sepsis Triage	78.5191

Role

	Predicted	Confidence
2	Role 2	81.0552

Time

	Predicted
2	16,593.0000

Label

Not Decided

1st Generative Historical Behaviour

	Activity	Role	Time
0	ER Registration	Role 2	11,106.0000
1	ER Triage	Role 5	24,928.0000

1stPrediction

Activity

	Predicted	Confidence
2	ER Sepsis Triage	78.5308

Role

	Predicted	Confidence
2	Role 2	81.0588

Time

	Predicted
2	16,596.0000

Label

Not Decided

2nd Prediction Historical Behaviour

	Activity	Role	Time
0	ER Registration	Role 2	0.0000
1	CRP	Role 4	24,928.0000

2ndPrediction

Activity

	Predicted	Confidence
2	Leucocytes	38.4078

Role

	Predicted	Confidence
2	Role 4	93.9186

Time

	Predicted
2	21,579.0000

Label

Not Decided

2nd Generative Historical Behaviour

	Activity	Role	Time
0	Leucocytes	Role 4	11,106.0000
1	CRP	Role 4	24,928.0000

2ndPrediction

Activity

	Predicted	Confidence
2	ER Registration	67.1770

Role

	Predicted	Confidence
2	Role 2	57.0298

Time

	Predicted
2	30,332.0000

Label

Not Decided

3rd Prediction Historical Behaviour

	Activity	Role	Time
0	ER Registration	Role 2	0.0000
1	Leucocytes	Role 2	24,928.0000

3rdPrediction

Activity

	Predicted	Confidence
2	ER Sepsis Triage	22.7034

Role

	Predicted	Confidence
2	Role 4	52.0468

Time

	Predicted
2	20,254.0000

Label

Not Decided

3rd Generative Historical Behaviour

	Activity	Role	Time
0	CRP	Role 5	11,106.0000
1	Leucocytes	Role 2	24,928.0000

3rdPrediction

Activity

	Predicted	Confidence
2	ER Registration	47.1102

Role

	Predicted	Confidence
2	Role 2	40.8165

Time

	Predicted
2	36,623.0000

Label

Not Decided

Figure 7.2: Recommendation on the dashboard for #1 Event

of this event. So, the decision will be based at the user's discretion. The user chose ER Sepsis Triage (5) as the action for this event which is in accordance with the process flow (Fig 3.2), and also the first recommendation. Thus, the prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5)>.

`#3 event:` After having executed ER Sepsis Triage (5), the predictive model recommended LacticAcid (9) with a confidence of 42.3%, CRP (3) with 20.9%, and IV Liquid (8) with 17.5%. None of the recommendations were of a particularly high confidence, being less than 50%, and neither were the one step ahead predictions of #2 event. Most of them predicted LacticAcid (9) to be one step ahead, but this does not agree with the process flow (Fig 3.2). However, our hypothesis 1 turned out to be true in the case of One-step ahead process outcome. IV Liquid (8) was taken as the action by the business user for this event which is following the process flow (Fig 3.2), and also the third recommendation. Considering the prefixes so far, the prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), IV Liquid (8)>.

`#4 event:` After having executed IV Liquid (8), the predictive model recommended CRP (3) with a confidence of 28.6%, LacticAcid (9) with 27.6% and IV Antibiotics (7) with 21.4%. The confidence is quite similar among the different predictive recommendations. Again, the user business acumen is required to decide the next course of action. However, most of the one-step ahead predictions such as Leucocytes (10), are also not as per the process flow (Fig 3.2), but it does satisfy our hypothesis 1 in the case of One-step ahead Process Outcome. IV Antibiotics (7) was chosen as the action by the business user for this event, which is per the process flow (Fig 3.2) and also the third recommendation. The chosen action being appended to the prefix for the next event would be, <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotics (7)>.

`#5 event:` After having executed IV Antibiotics (7), the recommended actions were LacticAcid (9) with a confidence of 33.0%, CRP (3) with 30.0%, and Leucocytes(10) with 23.3%. Nonetheless, once more the recommendations were not very confident. Thus, the user needs to use their knowledge about the process while selecting the next action. Most of the predictions of the One-step ahead predicted LacticAcid (9), which is not the most common behaviour of the process flow (Fig 3.2), but it satisfied our hypothesis 1 in the case of One-step ahead Process Outcome. Leucocytes (10) was picked as the next action, which is not as per the process flow (Fig 3.2), but the user might have selected keeping concurrency into account before admitting the patient to Admission NC. The action is the third predictive recommendation. It was also predicted as the one-step-ahead prediction in #4 event for process outcome which could have led to some intuition in the user to choose Leucocytes as an action. The resultant prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotics (7), Leucocytes (10)>.

`#6 event:` After having executed Leucocytes (10), the predictive model recommended Leucocytes (10) with a confidence of 38.6%, LacticAcid (9) with

30.4%, and CRP (3) with 39.9%, all with similar confidences. Also, the one-step-ahead prediction is not apparently clear-based on process flow (Fig 3.2). Although, both one-step ahead predictions satisfied the hypothesis 1, LacticAcid (9) was chosen as the next action which is as per the process flow (Fig 3.2) considering the action chosen in #5 event. It is also the second recommendation on the dashboard. It is important to note that the patient has not yet been admitted in Admission NC, although the user seems to incorporate concurrency in the process and the model has learnt that. Based on the selected action, the prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotics (7), Leucocytes (10), LacticAcid (9)>.

`#7 event`: After having executed LacticAcid (9), the recommendations for this event were Leucocytes (10) with a confidence of 31.6%, IV Antibiotics (7) with 22.9%, and CRP (3) with 16.0%, with no clear major confidence among them. However, the one-step-ahead predicted Admission NC (2) for both the cases in all the predictions which indicates that the patient needs to be admitted next and incorporates the concurrent behaviour. This will help in deciding the next action as it has strong confidence that the one step ahead action has to be Admission NC. Our hypothesis 1 strongly supports this event because the same prediction is predicted by all the predictions of both the One-step ahead. CRP (3) was picked as the next action and it was the third recommendation and also per the process flow (Fig 3.2) considering it comes after the action of #6 event. However, the patient admission being predicted in one-step ahead confirms that diagnostic actions (Leucocytes, LacticAcid and CRP) can be performed concurrently along with admitting the patient, which also helps in choosing the action. Based on the selected action, the prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotics (7), Leucocytes (10), LacticAcid (9), CRP (3)>.

`#8 event`: After having executed CRP (3), the recommendations provided for this event were Admission NC (2) with a confidence of 88.0%, IV Liquid (8) with 3.0%, and CRP (3) with the confidence of 2.9%. Admission NC seemed to be a suitable recommendation due to the high confidence of 88%, and this was also confirmed by one-step-ahead prediction of #7 event. One step ahead predicted the most prominent prediction, Leucocytes (10), which was based on the process flow, (Fig 3.2) considering it comes after Admission NC. This is again one of the circumstances where future predictions may help in selecting the action for the current event. Since Leucocytes (10) is being predicted as one-step ahead, and if the user applies their business knowledge based on process flow (Fig 3.2), then Admission NC seems to be the ideal action. Here again, both the one-step ahead predictions satisfy the hypothesis 1. Admission NC (2) was selected as the action, which was the first recommendation as well as a strong predictive outcome of #7 event. Although process flow (Fig 3.2) does not show this behaviour, it seems to be the case of concurrency where the patient is admitted while performing the diagnostics. The resulting next event prefixes would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotics (7), Leucocytes (10), LacticAcid (9), CRP (3), Admission NC (2)>.

`#9 event`: After having executed Admission NC (2), the recommendations for this event were Leucocytes (10) with a confidence of 38.8%, CRP (3) with 38.7%, and Admission NC (2) with 11.8%. The confidences of the first two recommendations are similar although, users need to decide what is best based on their business acumen. The one step ahead predicted Leucocytes (10) for all the predictions among the one-step ahead predictions, which might be a strong indicator to the user on what to select next. This could be concurrent to the one-step ahead predictive action. Once again, both the one-step ahead predictions strongly satisfied our hypothesis 1. CRP (3) was picked, which is in accordance with the process flow (Fig 3.2) and concurrent to the Leucocytes. It was the second recommendation on the dashboard. The resulting next event prefixes would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotics (7), Leucocytes (10), LacticAcid (9), CRP (3), Admission NC (2), CRP (3)>.

`#10 event`: After having executed CRP (3), the recommendations suggested by the predictive model were Leucocytes (10) with a confidence of 49.2%, CRP (3) with 34.5%, and Release A (11) with 10.9%. Leucocytes was predicted by all the one-step ahead in the #9 event, and the confidence of the recommendation was 49.2% ($\approx$ 50%). Furthermore, the one step ahead predicted Release A (11), which is an indication of process convergence towards the end of the case and is also based on the process flow (Fig 3.2). As anticipated, Leucocytes (10) was selected as the action, and it fits into the process flow as well, (Fig 3.2) considering CRP as the previous action and accounting for concurrency. The resulting next event prefixes would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotics (7), Leucocytes (10), LacticAcid (9), CRP (3), Admission NC (2), CRP (3), Leucocytes (10)>.

`#11 event`: After having executed Leucocytes (10), the recommendations proposed by the predictive models were Release A (11) with a confidence of 38.7%, Leucocytes (10) with 30.5%, and CRP (3) with 23.6%. Release A seemed to be the confident contender, considering it was also strongly predicted by the chosen recommendation in the earlier event one-step ahead prediction as well. The one-step-ahead prediction in this event perceived that the end of the case was approaching. Therefore, the most regular end of event activity Return ER (15) as per the process flow (Fig 3.2) was predicted. Thus, the next one-step ahead activity would not be performed in the next event. Release A (11) was the anticipated action which fits in the process flow (Fig 3.2) and also was the first recommendation. The resulting next event prefixes would be <ER Registration (4), ER Triage (6), ER

Sepsis Triage (5), IV Liquid (8), IV Antibiotics (7), Leucocytes (10), LacticAcid (9), CRP (3), Admission NC (2), CRP (3), Leucocytes (10), Release A (11)>.

`#12 event:` After having executed Release A (11), the predictive model recommended Return ER (15) with a confidence of 63.2%, end (17) with 35.7%, and Release A (11) with 0.3% out of which, Return ER had the highest confidence when compared to the other recommendations which appeared to be the most reasonable to take as next course of action and also per the process flow (Fig 3.2). Return ER (15) fits in the process flow (Fig 3.2) and also was the first recommendation. The resulting next event prefixes would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotics (7), Leucocytes (10), LacticAcid (9), CRP (3), Admission NC (2), CRP (3), Leucocytes (10), Release A (11), Return ER (15)>.

`#13 event:` After executing Return ER (15), dashboard marks the end of the trace of the "WFA" case-id.

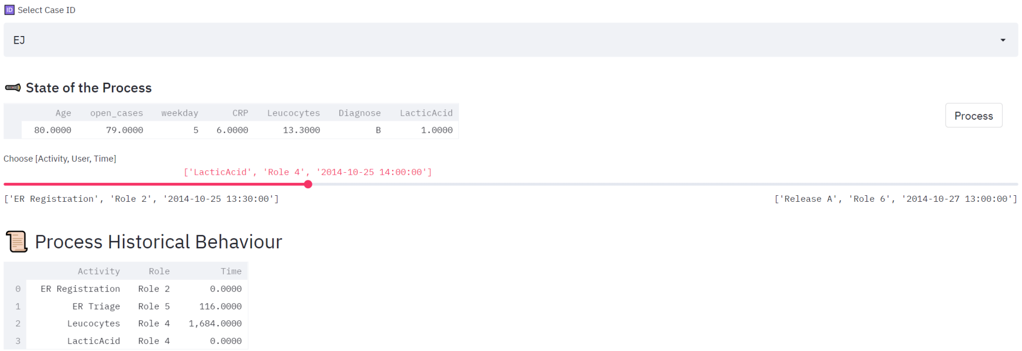
The explanations here do not account for the role and time. It also does not take into consideration the contextual information, i.e., inter and intra-case attributes (Table 3.2), which provides information about the state of the event that the user can consider while making decisions for the next action and allocate resource to it.

All in all, during the course of execution of "WFA" case-id, when the recommendation confidence of most probable outcome (1st Predicted Recommendation) was approximately greater than 50% it was selected as the next action, as demonstrated in events #0, #1, #2, #8, #10, and #12. However, when the confidence was below $\leq 50\%$, the user chose different actions from the most probable one, but those action choices were available on the dashboard, as demonstrated in events #3, #4, #5, #6, #7 and #9. The demonstration was considered successful as at each event, there was a recommendation that corresponded with the historical actions in the test log while replying.

7.2 Demonstration — Model Explainability

This section discusses the evaluation of the predictive model using the dashboard by replaying an executed case to assist the business user in determining the quality of the predicted recommendation using the Evaluation Mode. In this demonstration, we emulate how a business user might have interpreted the model's predictive quality from the information provided on the dashboard. The user could use it for similar cases to determine what the model would recommend and how cases would turn out if only the recommendations, from the dashboard for a similar situation, are chosen. Here, cases could be similar based on contextual information (inter- and intra-case) or cases that belong to the same variant. It tabulates the top three predictive recommendations (p_r) with their confidences (c). All activities are identified by their appropriate activity index values, comparable to what has been implemented in the Execution mode demonstration (Section 7.1).

We first select the pre-select prefixes of length four using the slider on the Evaluation mode which constitute the prefixes <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9)>, as shown in the Figure 7.6. This resulted in the generation of ‘End of Trace Generative’ (Subsection 5.2.1.2) and ‘End of Trace SME’ (Subsection 5.2.1.1), as shown in Table 7.2. It is essential to mention that case-id “EJ” is one of the variants in the Sepsis event log which doesn’t follow the most frequent path as shown in the process flow Figure 3.2.



Select Case ID

EJ

State of the Process

	Age	open_cases	weekday	CRP	Leucocytes	Diagnose	LacticAcid
	80.0000	79.0000	5	6.0000	13.3000	B	1.0000

Choose [Activity, User, Time]

['LacticAcid', 'Role 4', '2014-10-25 14:00:00']

['ER Registration', 'Role 2', '2014-10-25 13:30:00']

['Release A', 'Role 6', '2014-10-27 13:00:00']

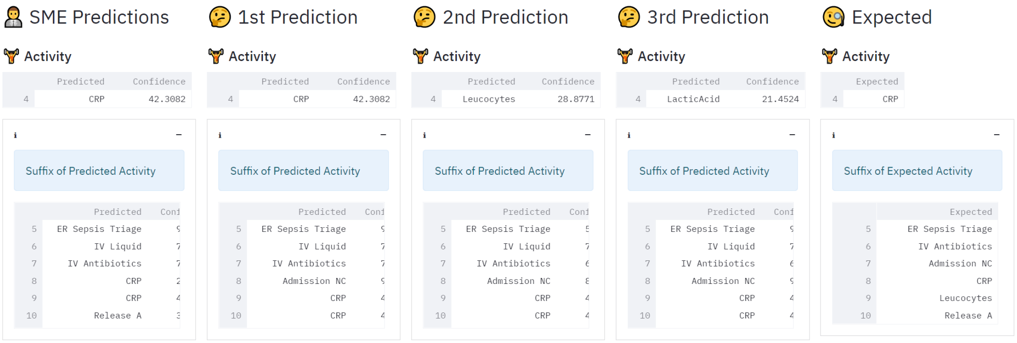
Process Historical Behaviour

	Activity	Role	Time
0	ER Registration	Role 2	0.0000
1	ER Triage	Role 5	116.0000
2	Leucocytes	Role 4	1,684.0000
3	LacticAcid	Role 4	0.0000

Figure 7.3: Pre-select prefix in the Evaluation Mode

Objective : The Evaluation Mode demonstration is regarded as successful when the End of the Trace of any Generative and the SME recommendations end with the event which indicates the end of case based on process flow as shown in the Figure 3.2). The activities that will be recognised as the end of the trace are Release A (11), Release B (12), Release C (13), Release D (14) and Return ER (15). If the end of trace does not end with any of these activities, the demonstration results are considered unsuccessful. The reason behind selecting this is that at least any of the End of the Trace Generative and End of Trace SME predictive recommendations would lead to the completion of a trace for a given case.

The Observed in Table 7.2 in the test log for the case-id “EJ” is meant for comparison and to highlight the ‘End of Trace Generative’ and ‘End of Trace SME’ when it matches. Also, the activity mentioned in the observation is what ’End of Trace SME‘ uses as a prefix, appending over the pre-select prefix to generate the most probable outcome until the end of the case, as shown in Figure 7.4, ‘SME Prediction’ is tabulated as ‘End of Trace SME’, the ‘1st prediction’, ‘2nd Prediction’, and ‘3rd Prediction’ are part of the ‘End of Trace Generative’, and the ‘Expected’ is the observed in Table 7.2.



SME Predictions

Activity

	Predicted	Confidence
4	CRP	42.3082

Suffix of Predicted Activity

	Predicted	Conf
5	ER Sepsis Triage	9
6	IV Liquid	7
7	IV Antibiotics	7
8	CRP	2
9	CRP	4
10	Release A	3

1st Prediction

Activity

	Predicted	Confidence
4	CRP	42.3082

Suffix of Predicted Activity

	Predicted	Conf
5	ER Sepsis Triage	9
6	IV Liquid	7
7	IV Antibiotics	7
8	Admission NC	9
9	CRP	4
10	CRP	4

2nd Prediction

Activity

	Predicted	Confidence
4	Leucocytes	28.8771

Suffix of Predicted Activity

	Predicted	Conf
5	ER Sepsis Triage	5
6	IV Liquid	7
7	IV Antibiotics	6
8	Admission NC	8
9	CRP	4
10	CRP	4

3rd Prediction

Activity

	Predicted	Confidence
4	LacticAcid	21.4524

Suffix of Predicted Activity

	Predicted	Conf
5	ER Sepsis Triage	9
6	IV Liquid	7
7	IV Antibiotics	6
8	Admission NC	9
9	CRP	4
10	CRP	4

Expected

Activity

	Expected
4	CRP

Suffix of Expected Activity

	Expected
5	ER Sepsis Triage
6	IV Antibiotics
7	Admission NC
8	CRP
9	Leucocytes
10	Release A

Figure 7.4: End of Case in the Evaluation Mode

Table 7.2: Case-id EJ End of Trace Evaluation

#event	Observed	End of Trace Generative						End of Trace SME	
		1st		2nd		3rd			
		p	c	p	c	p	c	p	c
0	0	-	-	-	-	-	-	-	-
1	4	-	-	-	-	-	-	-	-
2	6	-	-	-	-	-	-	-	-
3	10	-	-	-	-	-	-	-	-
4	9	**3**	**42.3**	10	28.9	9	21.5	**3**	**42.3**
5	3	**5**	**91.6**	**5**	**55.6**	**5**	**95.6**	**5**	**91.7**
6	5	8	79.7	8	71.5	8	72.2	8	79.7
7	7	7	79.0	7	67.2	7	69.2	7	78.9
8	2	2	98.6	2	84.6	2	98.1	**3**	**25.6**
9	3	3	45.5	3	45.4	3	46.8	3	49.5
10	10	3	46.3	3	47.1	3	46.8	**11**	**34.9**
11	11	-	-	-	-	-	-	-	-

1st, 2nd, 3rd : top three respective predictive recommendations.
p_r: predictive recommendation ; c: confidence (measured in %)
Activity Index — 1: Admission IC, 2: Admission NC, 3: CRP, 4: ER Registration, 5: ER Sepsis Triage, 6: ER Triage, 7: IV Antibiotics, 8: IV Liquid, 9: LacticAcid, 10: Leucocytes, 11: Release A, 12: Release B, 13: Release C, 14: Release D, 15: Return ER, 16: other, 0: start, 17: end

`#0 - #4 event:` Observed column of the events constitute the chosen pre-select prefix of the case (highlighted in green). Having executed <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9)> as the prefix, the model recommended CRP (3) with a confidence of 42.3%, LacticAcid (9) with 28.9% and Leucocytes (10) with 21.5% for the End of Trace Generative. In contrast, it recommended CRP (3) with a confidence of 42.3% for the End of Trace SME. The next action in the test log is CRP (3), which only matches the 1st End of Trace Generative and the End of Trace SME. However, considering the process flow (Fig. 3.2), LacticAcid (last element in the prefix) is not followed by CRP; rather it is the other way round. For the next event prediction, each prediction formed independent prefixes. So, the 1st End of Trace Generative prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3)>, the 2nd End of Trace Generative prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), LacticAcid (9)>, and the 3rd End of Trace Generative prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), Leucocytes (10)>. Similarly, the End of Trace SME prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3)>.

`#5 event:` After having executed prefixes for each of the predictions mentioned earlier, the dashboard showed ER Sepsis Triage (5) for the End of Trace Generative for all the recommendations with confidences of 91.6%, 55.6%, and 95.6% for 1st, 2nd, and 3rd recommendations, respectively. End of Trace SME also recommended ER Sepsis Triage (5) with a confidence of 91.7%. All of the recommendations matched with the observed next action in the test log, which is ER Sepsis Triage (5) but, considering the process flow (Fig. 3.2), CRP is not followed by ER Sepsis Triage and neither is it the other way round. For the next event, the 1st End of Trace Generative prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5)>, the 2nd End of Trace Generative prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), LacticAcid (9), ER Sepsis Triage (5)>, and the 3rd End of Trace Generative prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), Leucocytes (10), ER Sepsis Triage (5)>. Similarly, the End of Trace SME prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5)>.

`#6 event:` Following the execution of the prefixes mentioned in the previous event, the dashboard showed IV Liquid (8) for the End of Trace Generative for all the recommendations with confidences of 79.7%, 71.5%, and 72.2% for 1st, 2nd and 3rd recommendations, respectively. End of Trace SME also recommended IV Liquid (8) with a confidence of 79.7%. The recommendations did not match with the observed next action in the test log, which is IV Antibiotics (7), but considering the process flow (Fig. 3.2) ER Sepsis Triage is followed by IV Liquid and not with IV Antibiotics. Again, for the 1st End of Trace Generative, the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid

(9), CRP (3), ER Sepsis Triage (5), IV Liquid (8)>, for the 2nd End of Trace Generative the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), LacticAcid (9), ER Sepsis Triage (5), IV Liquid (8)>, and for the 3rd End of Trace Generative, the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), Leucocytes (10), ER Sepsis Triage (5), IV Liquid (8)>. Since the prefixes for the End of Trace SME comes from the test log, the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5), IV Antibiotic (7)>.

#7 event: After having executed prefixes for each of the predictions mentioned earlier, the dashboard showed IV Antibiotic (7) for the End of Trace Generative for all the recommendations, with confidences of 79.0%, 67.2%, and 69.2% for 1st, 2nd, and 3rd recommendations, respectively. End of Trace SME also recommended IV Antibiotic (7) with a confidence of 78.9%. The recommendations did not match with the observed next action in the test log, which is Admission NC (2), but considering the process flow (Fig. 3.2) IV Liquid is followed by IV Antibiotic and not with Admission NC. So, for the 1st End of Trace Generative, the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7)>. For the 2nd End of Trace Generative the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), LacticAcid (9), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7)> while for the 3rd End of Trace Generative the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), Leucocytes (10), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7)>. Since the prefixes for the End of Trace SME came from the test log, the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5), IV Antibiotic (7), Admission NC (2)>.

#8 event: Following the execution of prefixes mentioned in the previous event, the dashboard showed Admission NC (2) for the End of Trace Generative for all the recommendations with confidences of 98.6%, 84.6%, and 98.1% for 1st, 2nd, and 3rd recommendations, respectively. The End of Trace SME recommended CRP (3) with a confidence of 25.6%. The generative recommendations did not match with the observed next action in the test log, which is CRP (3), but the End of Trace SME did match although, considering the process flow, (Fig. 3.2) Admission NC is followed by CRP. Thus, End of Trace SME is also as per the process flow for this event. Again, for the 1st End of Trace Generative, the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7), Admission NC (2)>, for the 2nd End of Trace Generative the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), LacticAcid (9), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7), Admission NC (2)>, and for the 3rd End of Trace Generative the prefixes were <ER Registration (4), ER Triage (6), Leucocytes

(10), LacticAcid (9), Leucocytes (10), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7), Admission NC (2)>. For the End of Trace SME, the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5), IV Liquid (8), CRP (3)>.

`#9 event:` After having executed prefixes for each of the predictions mentioned above, the dashboard showed CRP (3) for the End of Trace Generative for all the recommendations with confidences of 45.5%, 45.4%, and 46.8% for 1st, 2nd, and 3rd recommendations, respectively. End of Trace SME also recommended CRP (3) with a confidence of 49.5%. The recommendations did not match the observed next action in the test log- Leucocytes (10). Now, considering the process flow, (Fig. 3.2) CRP is followed by Leucocytes as well as LacticAcid. However, all of the recommendations recommend that there is a self-loop on CRP, which is not noticed in the process flow (Fig. 3.2). So, for the 1st End of Trace Generative, the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7), Admission NC (2), CRP (3)>, for the 2nd End of Trace Generative the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), LacticAcid (9), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7), Admission NC (2), CRP (3)>, and for the 3rd End of Trace Generative the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), Leucocytes (10), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7), Admission NC (2), CRP (3)>. For the End of Trace SME, the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5), IV Liquid (8), CRP (3), Leucocytes (10)>.

`#10 event:` Following the execution of prefixes mentioned in the previous event, the dashboard showed CRP (3) again for the End of Trace Generative for all the recommendations with confidences of 46.3%, 47.1%, and 46.8% for 1st, 2nd, and 3rd recommendations, respectively. End of Trace SME recommended Release A (11) with a confidence of 34.9%. The generative recommendations did not match with the observed next action in the test log, which is Release A (11), but the End of Trace SME did match. So, considering the process flow (Fig. 3.2), CRP is followed by Release A. Thus, End of Trace SME is also as per the process flow for this event. Therefore, for the 1st End of Trace Generative the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7), Admission NC (2), CRP (3), CRP (3)>, for the 2nd End of Trace Generative the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), LacticAcid (9), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7), Admission NC (2), CRP (3), CRP (3)>, and for the 3rd End of Trace Generative the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), Leucocytes (10), ER Sepsis Triage (5), IV Liquid (8), IV Antibiotic (7), Admission NC (2), CRP

(3), CRP (3)>. For the End of Trace SME the prefixes were <ER Registration (4), ER Triage (6), Leucocytes (10), LacticAcid (9), CRP (3), ER Sepsis Triage (5), IV Liquid (8), CRP (3), Leucocytes (10), Release A (11)>.

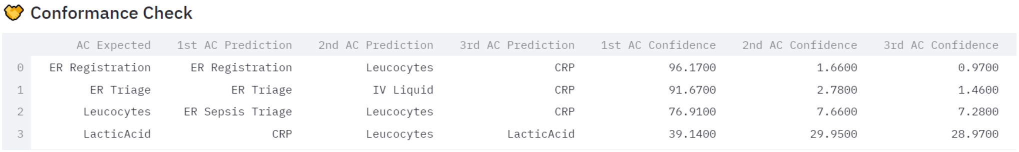
`#11 event`: This event marks the end of the case "EJ" as the Release A (11) was also observed the length of the trace of case "EJ" up until here. Thus, the predictive recommendations did not recommend the observed behaviour,

For events `event #5 to #9`, the predictive recommendations appear to have learnt the most frequent behaviour, which certainly matches with the process flow in Figure 3.2. Case-Id "EJ" seems to follow a deviant path when the Observed column behaviour is compared with the process flow (Fig. 3.2). The first two events follow the frequent behaviour which is ER Registration (4) and ER Triage (6). Then, Leucocytes (10) is observed at event #3, which is not observed in the process flow (Fig. 3.2) after ER Triage, LacticAcid (9) is observed after Leucocytes in event #4 which is observed in the process flow (Fig. 3.2). Next, CRP (3) is observed which is again not observed in the process flow (Fig. 3.2) at event #5 following which, ER Sepsis Triage (5) is observed which is also not in the process flow (Fig. 3.2). IV Antibiotics (7) is observed at event #7, which matches with the process flow (Fig. 3.2) after ER Sepsis Triage. Next, Admission NC (2) is observed at event #8 which is also as per the process flow (Fig. 3.2) after IV Antibiotics. Following it, CRP (3) is observed at event #9 which is as per the process flow (Fig. 3.2) after Admission NC. Next, Leucocytes (10) is observed, which is also in the process flow (Fig. 3.2) after CRP. Lastly, Release A is observed at event #11, which is not present in the process flow (Fig. 3.2) after Leucocytes. Thus, it seems that the observed behaviour in the test log has the influence of contextual information, and there is a concurrency among the case-id "EJ" variant, which is not observed in the process flow (Fig. 3.2).

Nonetheless, generative predictive behaviour is generalizable as can be seen in events `event #5 to #9`, but when accounting for similar cases, the inter- and intra-case features might have caused the actions in the case-id "EJ". Users need to be extra cautious and rely more on SME predictions (End of Trace SME) while making decisions. Thus, the Execution mode recommendations will be helpful. Also, throughout the `event #5 to #10` our hypothesis 1 was strongly supported, which also explains the similarity values in the Table 6.4 in the Section 6.4.

After performing the End of Trace predictions, the dashboard showed the conformance check (Subsection 5.2.1.3) at the end in a tabular form for case-id "EJ", as shown in the Figure 7.5.

A conformance check is performed over the pre-select prefix to witness how much the model knows about the pre-select prefix. In the conformance check of the first four pre-select prefixes of case-id "EJ", tabulated in Table 7.3, it was observed that the first predictive recommendation advised the first two prefixes correctly, while the second predictive recommendation recommended the third prefix. Lastly,



Conformance Check

	AC Expected	1st AC Prediction	2nd AC Prediction	3rd AC Prediction	1st AC Confidence	2nd AC Confidence	3rd AC Confidence
0	ER Registration	ER Registration	Leucocytes	CRP	96.1700	1.6600	0.9700
1	ER Triage	ER Triage	IV Liquid	CRP	91.6700	2.7800	1.4600
2	Leucocytes	ER Sepsis Triage	Leucocytes	CRP	76.9100	7.6600	7.2800
3	LacticAcid	CRP	Leucocytes	LacticAcid	39.1400	29.9500	28.9700

Figure 7.5: Conformance Check in the Evaluation Mode

the third predictive recommendation predicted the fourth prefix. Although, as mentioned earlier, pre-select prefixes did not follow the frequent process flow (Fig. 3.2), this suggests that relying on the maximum confidence prediction may not be accurate every time and providing the user with other recommended actions will allow them to choose an ideal action for the scenario. However, this observation is based on this and a few other case-id tested during the course of the experiment.

Table 7.3: Pre-Select Prefixes of Case-id EJ and the Conformance Check

#event	Pre-select Prefix	Conformance Check					
		1st		2nd		3rd	
		p_r	c	p_r	c	p_r	c
0	0	**4**	**96.2**	10	1.7	3	1.0
1	4	**6**	**91.7**	8	2.8	3	1.5
2	6	5	77.0	**10**	**7.7**	3	7.3
3	10	3	39.1	10	30.0	**9**	**29.0**
4	9	**-**	**-**	-	-	-	-

1st, 2nd, 3rd : top three respective predictive recommendations.
p_r: predictive recommendation ; c: confidence (measured in %)
Activity Index — 1: Admission IC, 2: Admission NC, 3: CRP, 4: ER Registration, 5: ER Sepsis Triage, 6: ER Triage, 7: IV Antibiotics, 8: IV Liquid, 9: LacticAcid, 10: Leucocytes, 11: Release A, 12: Release B, 13: Release C, 14: Release D, 15: Return ER, 16: other, 0: start, 17: end

The demonstration explanation excludes the role, as these need resource planning and business knowledge. It also ignores the inter- and intra-case characteristics, assuming that the user will apply explainability to the assessment while looking at these attributes during runtime. The MAE time comparison of the Observed, 1st End of Trace Generative, 2nd End of Trace Generative, 3rd End of Trace Generative, and End of Trace SME. All the generative predictions followed the same pattern of MAE with slight variations. End of Trace SME follows the same pattern as generative until event #5, #6, and #7. Following this, at event #8, it has a higher

MAE than the generative, at event #9, it has a lower MAE, and at event #10, the MAE shoots up compared to generative. The difference between both the End of Trace and the observed is quite large. The average of the MAEs is tabulated in Table 7.4. The observed MAE is 0.33 while the 1st End of Trace Generative, 2nd End of Trace Generative, and 3rd End of Trace Generative's MAE are 0.93, 0.95, and 0.99, respectively. End of Trace SME has the highest MAE among all with a value of 1.02. Thus, in terms of waiting time, all of the predictions performed similarly.

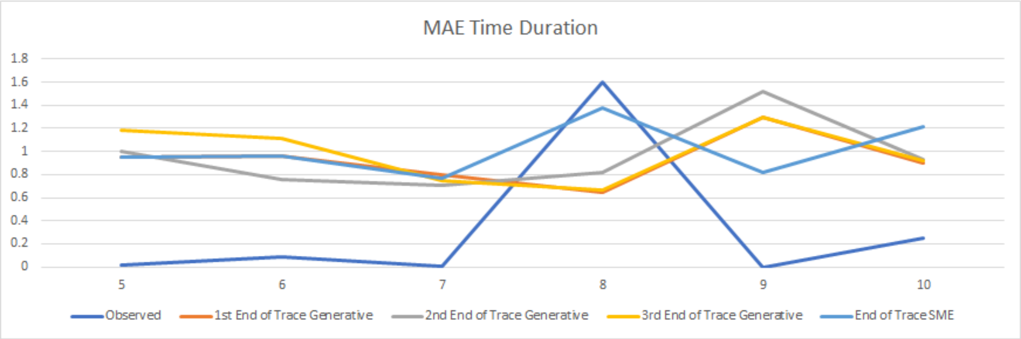


Figure 7.6: Time Duration MAE Comparison (In Days)

Table 7.4: Average MAE for the case-id 'EJ' for the event #5 to #10

Observed	End of Trace Generative			End of Trace SME
	1st End of Trace Generative	2nd End of Trace Generative	3rd End of Trace Generative	
0.33	0.93	0.95	0.99	1.02

The entire demonstration was partially successful because the End of Trace SME ended with Release A (11), while none of the End of Trace Generative ended with the activity which marks the end of the trace as per the process flow (Fig. 3.2).

7.3 Demonstration — What-If Scenarios

This section focuses on the What-If dashboard capabilities using historical traces from the test log. We will look into two case id's — DL and VD, and try to steer both of them with each other's observed behaviour as much as possible. The contextual information of DL and VD is tabulated in Table 7.5, and they are relatively similar in terms of Age, open-cases and LacticAcid.

The demonstration aims to look at the dashboard's capabilities, whether it allows completing the trace with a similar sequence of activities for two cases that are comparable in terms of contextual information, and able to converge towards the end with the same activity of each other. Concerning the Sepsis cases :

Table 7.5: Contextual Information of Case DL AND VD

Case-Id	Age	Open Cases	LacticAcid	CRP	Leucocytes	Diagnose
DL	60	44 – 48	0 – 1.7	0 – 48	0 – 2.2	JB
VD	50	43 – 47	0 – 1.1	0 – 78	0 – 6.7	H

I `Diagnostic` actions (LacticAcid (9), Leucocytes (10) and CRP (3)) can be performed in any order (concurrency) and at any step once registration (ER Registration (4)) of the patient is completed. The concurrency is shown in Appendix Figure A.3.

II Patient `medication` can be interchanged. According to the process flow (Fig. 3.2), IV Liquid (8) is followed by IV Antibiotics (7), although the other way round is also possible as shown in Appendix Figure A.1 of the event log.

III The `urgency` of case (ER Triage (6), ER Sepsis Triage (5)) directly follows each other, which is evident from the process flow (Fig. 3.2), while there are few cases when ER Sepsis Triage is followed by ER Triage, as shown in Appendix Figure A.2.

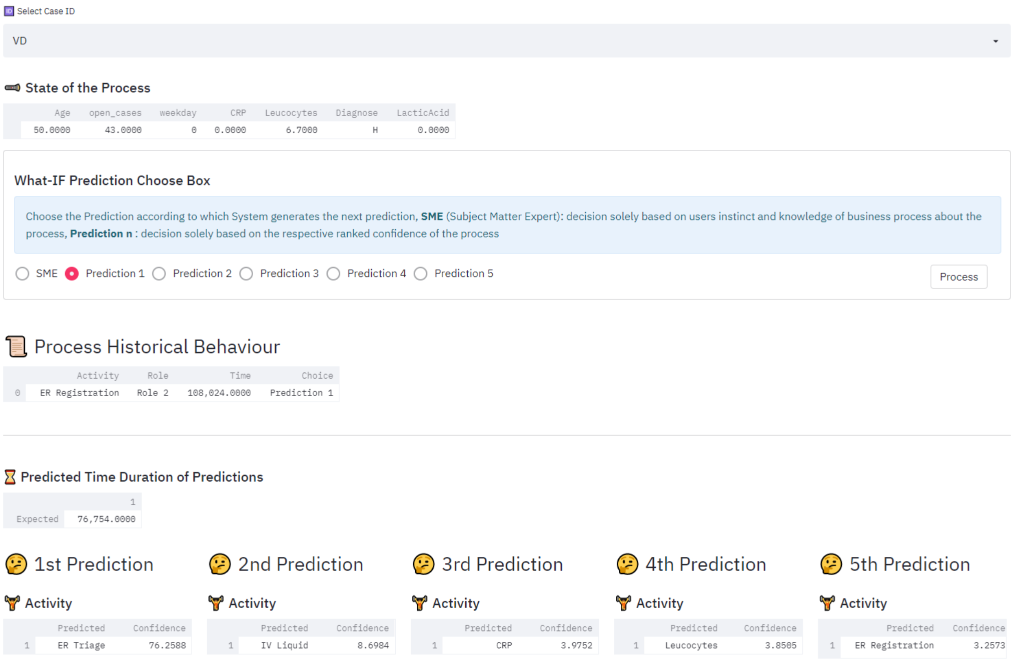
Objective : Considering the above points (I, II, III), if the demonstration can steer the two historical traces with similar contextual information, which could converge them to the same end activity, it would be considered successful. This will help the user to form a local post-hoc explanation of similar case outcomes when the action is taken on the ongoing case. Table 7.6 tabulates the activity sequence of case-id DL and VD, which is in the test log. Thus, based on our objective, if the projection of each case can be performed following a similar sequence along with the concurrent situations, and if it causes both the cases to converge to Release A (11), the demonstration would be considered successful.

Table 7.7 tabulates the closest projection of case-id DL on VD, while Table 7.8 does the vice-versa. The column ‘Case-Id DL Events’ reflects the sequence of events played out for case-id DL in the test log, and similarly ‘Case-Id VD Events’ for case-id VD. The ‘Chosen Recommendation’ is tabulated after finding the sequence of choices that closely follow each case’s projection, following the concurrent behaviour discussed earlier. The entire demonstration experiment was run multiple times to try different choices selected at each event to get the closest matching execution sequence on each other in one go, and it has been discussed wherever the LSTM model struggled in subsection 7.3.1 and 7.3.2. Figure 7.7 shows the dashboard at the event #1 situation, which is tabulated in Table 7.7 for the case-id VD.

Table 7.6: Sequence of events for case-id VD AND DL

Case-Id	Event Sequence										
	1	2	3	4	5	6	7	8	9	10	11
DL	4	6	5	10	3	9	8	7	2	10	11
VD	4	10	9	3	6	5	8	2	2	7	11

Activity Index — 1: Admission IC, 2: Admission NC, 3: CRP, 4: ER Registration, 5: ER Sepsis Triage, 6: ER Triage, 7: IV Antibiotics, 8: IV Liquid, 9: LacticAcid, 10: Leucocytes, 11: Release A, 12: Release B, 13: Release C, 14: Release D, 15: Return ER, 16: other, 0: start, 17: end



Select Case ID

VD

State of the Process

	Age	open_cases	weekday	CRP	Leucocytes	Diagnose	LacticAcid
	50.0000	43.0000	0	0.0000	6.7000	H	0.0000

What-IF Prediction Choose Box

Choose the Prediction according to which System generates the next prediction, **SME** (Subject Matter Expert): decision solely based on users instinct and knowledge of business process about the process, **Prediction n** : decision solely based on the respective ranked confidence of the process

○ SME ● Prediction 1 ○ Prediction 2 ○ Prediction 3 ○ Prediction 4 ○ Prediction 5

Process Historical Behaviour

	Activity	Role	Time	Choice
0	ER Registration	Role 2	108,024.0000	Prediction 1

Predicted Time Duration of Predictions

	1
Expected	76,754.0000

1st Prediction

Activity

	Predicted	Confidence
1	ER Triage	76.2588

2nd Prediction

Activity

	Predicted	Confidence
1	IV Liquid	8.6984

3rd Prediction

Activity

	Predicted	Confidence
1	CRP	3.9752

4th Prediction

Activity

	Predicted	Confidence
1	Leucocytes	3.8505

5th Prediction

Activity

	Predicted	Confidence
1	ER Registration	3.2573

Figure 7.7: What-If of Case-id 'VD' for the event #1

7.3.1 Case-id DL projection on VD

The following findings are based on Table 7.7 :

`#0 event`: The first recommendation started with zero-prefix, and it recommended ER Registration (4) with a confidence of 97.2%, Leucocytes (10) with 0.7%, CRP (3) with 0.6%, IV Liquid (8) with 0.5%, and ER Sepsis Triage (5) with 0.4%. Out of these, ER Registration had the highest confidence at 97.2%. The user might choose ER Registration (4) as the next action for this event, and this is

Table 7.7: What-If Simulation of Case-Id VD

#event	Case-Id DL Events	Chosen Recommendation	Predicted Recommendation									
			1st		2nd		3rd		4th		5th	
			p_r	c	p_r	c	p_r	c	p_r	c	p_r	c
0	0	0	**4**	**97.2**	10	0.7	3	0.6	8	0.5	5	0.4
1	4	4	**6**	**76.3**	8	8.7	3	4.0	10	3.9	4	3.3
2	6	6	**5**	**77.8**	10	10.0	9	5.1	3	4.9	8	1.6
3	5	5	**10**	**36.4**	9	32.7	3	23.7	8	2.8	5	2.3
4	3	10	**9**	**48.7**	3	24.8	10	17.3	5	4.6	6	3.0
5	9	9	5	56.2	9	19.6	**3**	**7.2**	8	6.2	10	5.8
6	10	3	**8**	**51.0**	5	36.7	7	7.8	2	1.6	9	1.2
7	7	8	**7**	**45.9**	2	38.2	1	12.1	8	3.1	11	0.3
8	8	7	**2**	**48.0**	1	13.6	3	12.0	10	10.1	11	6.8
9	2	2	3	40.3	**10**	**33.3**	2	10.3	9	7.1	11	6.4
10	10	10	3	44.8	10	28.6	2	11.4	**11**	**7.4**	9	4.2
11	11	11	-	-	-	-	-	-	-	-	-	-

1st, 2nd, 3rd, 4th, 5th : top five predictive recommendations.

p_r: predictive recommendation ; c: confidence (measured in %)

Activity Index — 1: Admission IC, 2: Admission NC, 3: CRP, 4: ER Registration, 5: ER Sepsis Triage, 6: ER Triage, 7: IV Antibiotics, 8: IV Liquid, 9: LacticAcid, 10: Leucocytes, 11: Release A, 12: Release B, 13: Release C, 14: Release D, 15: Return ER, 16: other, 0: start, 17: end

identical with event #1 of case-id DL. The choice was appended to the prefix of length zero.

`#1 event:` After having executed ER Registration, the dashboard recommended ER Triage (6) with a confidence of 76.3%, IV Liquid (8) with 8.7%, CRP (3) with 4.0%, Leucocytes (10) with 3.9%, and ER Registration (4) with 3.3%. Out of these, ER Triage had the highest confidence at 76.3%. The user might choose ER Triage (6) as the next action for this event owing to its high confidence, and this is identical with event #2 of case-id DL. The prefix for the next event would be <ER Registration (4), ER Triage (6)>.

`#2 event:` After executing ER Triage, the dashboard recommended ER Sepsis Triage (5) with a confidence of 77.8%, Leucocytes (10) with 10.0%, LacticAcid (9) with 5.1%, CRP (3) with 4.9%, and IV Liquid (8) with 1.6%. Out of these, ER Sepsis Triage had the highest confidence at 77.8%. The user might choose ER Sepsis Triage (5) as the next action for this event, and this is identical with event #3 of case-id DL. The prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5)>.

`#3 event:` After executing ER Sepsis Triage, the dashboard recommended Leucocytes (10) with a confidence of 36.4%, LacticAcid (9) with 32.7%, CRP (3) with 23.7%, IV Liquid (8) with 2.8%, and ER Sepsis Triage (5) with 2.3%. As the confidences are not very strong for any of them and the first three recommendations are related to diagnostics, concurrent behaviour among them can be incorporated (I). So, Leucocytes (10) might be selected and this matches with the event #6 of case-id DL. The prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), Leucocytes (10)>.

`#4 event:` After executing Leucocytes, the dashboard recommended LacticAcid (9) with a confidence of 48.7%, CRP (3) with 24.8%, Leucocytes (10) with 17.3%, ER Sepsis Triage (5) with 4.6%, and ER Triage (6) with 3.0%. Out of these, LacticAcid had the highest confidence at 48.7% which is close to 50%. The user might select LacticAcid (9) and this matches with the event #5 of case-id DL. The prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), Leucocytes (10), LacticAcid (9)>.

`#5 event:` After executing LacticAcid, the dashboard recommended ER Sepsis Triage (5) with a confidence of 56.2%, LacticAcid (9) with 19.6%, CRP (3) with 7.2%, IV Liquid (8) with 6.2%, and Leucocytes (10) with 5.8%. The diagnostic (I) action cycle of the case has to be completed; thus, CRP (3) was chosen. It matches with event #4 of case-id DL. The prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), Leucocytes (10), LacticAcid (9), CRP (3)>.

`#6 event:` After executing CRP, the dashboard recommended IV Liquid (8) with a confidence of 51.0%, ER Sepsis Triage (5) with 36.7%, IV Antibiotics (7) with 7.8%, Admission NC (2) with 1.6%, and LacticAcid (9) with 1.2%. Out of these, IV Liquid had the highest confidence at 51.0%. Users might select IV Liquid (8) because it is historically chosen, has high confidence, and is identical to event #8 of case-id DL. The prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), Leucocytes (10), LacticAcid (9), CRP (3), IV Liquid (8)>.

`#7 event:` After executing IV Liquid, the dashboard recommended IV Antibiotic (7) with a confidence of 45.9%, Admission NC (2) with 38.2%, Admission IC (1) with 12.1%, IV Liquid (8) with 3.1%, and Release A (11) with 0.3%. Out of these, IV Antibiotic had the highest confidence at 45.9%, and it has been historically chosen after IV Liquid (8). So, the user might select IV Antibiotic (7) and it is also identical to event #7 of case-id DL. The prefix for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), Leucocytes (10), LacticAcid (9), CRP (3), IV Liquid (8), IV Antibiotic (7)>.

`#8 event:` After executing IV Antibiotic, the dashboard recommended Admission NC (2) with a confidence of 48.0%, Admission IC (1) with 13.6%, CRP (3) with 12.0%, Leucocytes (10) with 10.1%, and Release A (11) with 6.8%. Out of these, Admission NC had the highest confidence at 48.0% which is close to 50%. The user might select Admission NC, which is also historically chosen. It is identical with event #9 of case-id DL. The prefix for the next event would be <ER

Registration (4), ER Triage (6), ER Sepsis Triage (5), Leucocytes (10), LacticAcid (9), CRP (3), IV Liquid (8), IV Antibiotic (7), Admission NC (2)>.

`#9 event:` After executing Admission NC, the dashboard recommended CRP (3) with a confidence of 40.3%, Leucocytes (10) with 33.3%, Admission NC (2) with 10.3%, LacticAcid (9) with 7.1%, and Release A (11) with 6.4%. To converge along the case-id DL as this option is available on the dashboard, the user might select Leucocytes which is identical to event #10 of case-id DL. The prefixes for the next event would be <ER Registration (4), ER Triage (6), ER Sepsis Triage (5), Leucocytes (10), LacticAcid (9), CRP (3), IV Liquid (8), IV Antibiotic (7), Admission NC (2), Leucocytes (10)>.

`#10 - #11 event:` After executing Leucocytes, the dashboard recommended CRP (3) with a confidence of 44.8%, Leucocytes (10) with 28.6%, Admission NC (2) with 11.4%, Release A (11) with 7.4%, and LacticAcid (9) with 4.2%. To converge the trace to the end, the user might select Release A, which is also historically chosen and is identical to event #11 of case-id DL. This would mark the end of the trace for case-id VD.

This demonstration was successful as the trace of case-id VD was stirred successfully using the concurrent behaviour of diagnostic (I) and medication (II) actions and the end activity Release A was identical to the case-id DL as well.

LSTM Model Behaviour for Cas-Id VD : During the experiment, the different sequences of the diagnostic (I) actions were tried multiple times in various sequences. LSTM struggled to provide the most resembling actions in terms of concurrency. For instance, if CRP (3) was selected in event #3, executing that led to having one of the recommendations as LacticAcid (9). After executing LacticAcid, there were no recommendations for Leucocytes (10) even though it is present in the process flow (Fig. 3.2). Thus, considering the diagnostic actions to be concurrent (I) to each other lead us to performing the entire experiment multiple times in order to achieve all three diagnostic activities directly following each other in any sequence. Another challenge we faced was that not all the diagnostic sequences of actions were recommending either of the medication (II) recommendations at event #6. To tackle this, we re-ran the experiment to achieve the medication recommendation at event #6. Lastly, choosing the medication recommendation IV Antibiotic (7) at event #6 was causing IV Antibiotic (8) to not be recommended for any of the predictions. This led us to choose IV Liquid (8) first and then followed by IV Antibiotic (7), which is as per the process flow (Fig. 3.2).

7.3.2 Case-id VD projection on DL

The following points were derived from Table 7.8 :

Table 7.8: What-If Simulation of Case-Id DL

#event	Case-Id VD Events	Chosen Recommendation	Predicted Recommendation									
			1st		2nd		3rd		4th		5th	
			p_r	c	p_r	c	p_r	c	p_r	c	p_r	c
0	0	0	**4**	**97.2**	10	0.7	3	0.6	8	0.5	5	0.4
1	4	4	6	88.6	8	4.9	3	1.9	4	1.5	**10**	**1.2**
2	10	10	5	81.3	10	6.0	8	4.4	**3**	**3.3**	9	3.2
3	9	3	8	37.3	10	25.2	**9**	**15.3**	3	13.0	5	5.1
4	3	9	9	29.6	**5**	**29.5**	10	20.5	3	16.6	8	1.7
5	6	5	9	43.8	**6**	**14.0**	5	13.9	10	13.2	3	11.6
6	5	6	5	86.2	9	3.5	**8**	**3.4**	6	2.6	10	1.4
7	7	8	7	48.4	**2**	**40.7**	8	6.0	1	2.3	5	0.9
8	2	2	2	57.7	10	11.1	3	8.8	1	8.1	**7**	**4.1**
9	2	7	3	36.9	10	26.6	**2**	**13.7**	17	10.7	9	3.8
10	8	2	3	42.2	10	40.4	9	7.0	2	7.0	**11**	**2.2**
11	11	11	-	-	-	-	-	-	-	-	-	-

1st, 2nd, 3rd, 4th, 5th : top five predictive recommendations.
p_r: predictive recommendation ; c: confidence (measured in %)
Activity Index — 1: Admission IC, 2: Admission NC, 3: CRP, 4: ER Registration, 5: ER Sepsis Triage, 6: ER Triage, 7: IV Antibiotics, 8: IV Liquid, 9: LacticAcid, 10: Leucocytes, 11: Release A, 12: Release B, 13: Release C, 14: Release D, 15: Return ER, 16: other, 0: start, 17: end

`#0 event:` The first recommendation started with zero-prefix, and it recommended ER Registration (4) with a confidence of 97.2%, Leucocytes (10) with of 0.7%, CRP (3) with 0.6%, IV Liquid (8) with 0.5%, and ER Sepsis Triage (5) with 0.4%. Out of these, ER Registration had the highest confidence at 97.2%. The user might choose ER Registration (4) as the next action for this event, and this is identical with event #1 of case-id VD. The choice was appended to the prefix of length zero.

`#1 event:` After having executed ER Registration, the dashboard recommended ER Triage (6) with a confidence of 88.6%, IV Liquid (8) with 4.9%, CRP (3) with 1.9%, ER Registration (4) with 1.5%, and Leucocytes (10) with 1.2%. The user might select Leucocytes to follow along the case-id VD in event #2. The prefixes for the next event would be <ER Registration (4), Leucocytes (10)>.

`#2 event:` After executing Leucocytes, the dashboard recommended ER Sepsis Triage (5) with a confidence of 81.3%, Leucocytes (10) with 6.0%, IV Liquid (8) with 4.4%, CRP (3) with 3.3%, and LacticAcid (9) with 3.2%. To follow along the diagnostic (I) activities, the most confident diagnostic action could be selected by the user, which is CRP, and this is identical to event #4 of case-id VD. The prefixes for the next event would be <ER Registration (4), Leucocytes (10), CRP (3)>.

#3 event: After executing CRP, the dashboard recommended IV Liquid (8) with a confidence of 37.3%, Leucocytes (10) with 25.2%, LacticAcid (9) with 15.3%, CRP (3) with 13.0%, and ER Sepsis Triage (5) with 5.1%. As the confidences are not very strong for any of them, and the concurrency of diagnostic activities (I) have to be incorporated, the user might select LacticAcid, which is identical to event #3 of case-id VD. The prefixes for the next event would be <ER Registration (4), Leucocytes (10), CRP (3), LacticAcid (9)>.

#4 event: After executing LacticAcid, the dashboard recommended LacticAcid (9) with a confidence of 29.6%, ER Sepsis Triage (5) with 29.5%, Leucocytes (10) with 20.5%, CRP (3) with 16.6%, and IV Liquid (8) with 1.7%. The confidences are again not very strong for anyone of them. Although, based on process history, if the urgency is not evaluated just after registration (ER Registration), it is then followed with diagnosis. On applying the urgency concurrent behaviour (III) the user might select ER Sepsis Triage as it is the only recommendation available for determining urgency and this is identical to event #6 of case-id VD. The prefixes for the next event would be <ER Registration (4), Leucocytes (10), CRP (3), LacticAcid (9), ER Sepsis Triage (5)>.

#5 event: After executing ER Sepsis Triage, the dashboard recommended LacticAcid (9) with a confidence of 43.8%, ER Triage (6) with 14.0%, ER Sepsis Triage (5) with 13.9%, Leucocytes (10) with 13.2%, and CRP (3) with 11.6%. Again, the confidences are not very strong for any of them. Based on process history and to incorporate the urgency sequence (III) behaviour, the user might select ER Triage, and this is identical to event #5 of case-id VD. The prefixes for the next event would be <ER Registration (4), Leucocytes (10), CRP (3), LacticAcid (9), ER Sepsis Triage (5), ER Triage (6)>.

#6 event: After executing ER Triage, the dashboard recommended ER Sepsis Triage (5) with a confidence of 86.2%, LacticAcid (9) with 3.5%, IV Liquid (8) with 3.4%, ER Triage (6) with 2.6%, and Leucocytes (10) with 1.4%.Although ER Sepsis Triage had the highest confidence, it might not be chosen because the determination of urgency has already occurred in the trace. Based on the next event of case-id VD, on applying the medication concurrency to the activity IV Antibiotic which belongs to the medication group of actions (III), the user might choose IV Liquid. This is valid from historically chosen activities point of view as IV Liquid is followed by IV Antibiotics in the process flow (Fig. 3.2). The prefixes for the next event would be <ER Registration (4), Leucocytes (10), CRP (3), LacticAcid (9), ER Sepsis Triage (5), ER Triage (6), IV Liquid (8)>.

#7 event: After executing IV Liquid, the dashboard recommended IV Antibiotics (7) with a confidence of 48.4%, Admission NC (2) with 40.7%, IV Liquid (8) with 6.0%, Admission IC (1) with 2.3%, and ER Sepsis Triage (5) with 0.9%. Although IV Antibiotics showed confidence close to 50%, the user might want to stir the trace close to case-id VD. Thus, the user might end up selecting Admission

NC, and this is identical to the immediate next event #8 of case-id VD. The prefixes for the next event would be <ER Registration (4), Leucocytes (10), CRP (3), LacticAcid (9), ER Sepsis Triage (5), ER Triage (6), IV Liquid (8), Admission NC (2)>.

`#8 event:` After executing Admission NC, the dashboard recommended Admission NC (2) with a confidence of 57.7%, Leucocytes (10) with 11.1%, CRP (3) with 8.8%, Admission IC (1) with 8.1%, and IV Antibiotic (7) with 4.1%. After admission into the care unit (Admission NC), the rest of the medications can be completed. Thus, the user might select IV Antibiotic (7) and this is identical to event #10 of case-id VD. The prefixes for the next event would be <ER Registration (4), Leucocytes (10), CRP (3), LacticAcid (9), ER Sepsis Triage (5), ER Triage (6), IV Liquid (8), Admission NC (2), IV Antibiotic (7)>.

`#9 event:` After executing IV Antibiotic, the dashboard recommended CRP (3) with a confidence of 36.9%, Leucocytes (10) with 26.6%, Admission NC (2) with 13.7%, end (17) with 10.7%, and LacticAcid (9) with 3.8%. However, to stir the case converging towards the case-id VD, the user might select Admission NC and this is identical to event #9 of case-id VD. The resulting prefixes for the next event would be <ER Registration (4), Leucocytes (10), CRP (3), LacticAcid (9), ER Sepsis Triage (5), ER Triage (6), IV Liquid (8), Admission NC (2), IV Antibiotic (7), Admission NC (2)>.

`#10 - #11 event:` After executing Admission NC, the dashboard recommended CRP (3) with a confidence of 42.2%, Leucocytes (10) with 40.8%, LacticAcid(9) with 7.0%, Admission NC (2) with 7.0% and Release A (11) with 2.2%. To converge the trace to the end, the user might select Release A, which is also historically chosen and is identical to event #11 of case-id VD. This would mark the end of trace for case-id DL.

This demonstration is successful as the trace of case-id DL was stirred successfully using the concurrent behaviour of diagnostic (I), medication (II), and urgency (III) actions and also, the end activity Release A was identical to the case-id VD.

LSTM Model Behaviour for Cas-Id DL : During the experiment, the different sequences of the diagnostic (I) actions were tried multiple times in various sequences, just as we did for case-id VD. LSTM struggled to provide the possible next recommendation actions after performing the diagnostic actions in the order it was in case-id VD. In the instance of case-id DL, at event #1, the option to select Leucocytes (10) was available, so we selected it. For event #2 and event #3, on the other hand, the selections were interchanged as there were no recommendations related to urgency if the diagnostic activity were executed in the same sequence as event #4. Thus, the concurrency of diagnostic events played out to get the recommendation. LSTM did not learn cases where urgency could be determined after diagnostics effectively as there were very few examples in the Sepsis Event Log although, it did recommend ER Sepsis Triage (5) at event #4. Again at event #6, the only medication action available was IV Liquid (8); thus, it was selected as the next event. The two subsequent instances of Admission NC (2) in Case-id VD at event #8 and #9 was also learnt by the model, as there are approximately 15% cases in the event log where this happened. The model learnt this as it recommended Admission NC (2) as the most confident recommendation at event #9. However, there are only very fewer traces in the log where Admission NC was followed by any of the medication actions because of which, LSTM struggled to recommend any of the medication action after selecting Admission NC (2) two times in a row. So, we played out the experiment multiple times to get the sequence of action where Admission NC (2) was selected at event #7 followed by medication action IV Antibiotics (7) at event #8 and then the selected Admission NC (2) again at event #9.

Again, the demonstration explanation excludes the role and time, as these need resource planning and business knowledge. It also ignores the contextual information (inter- and intra-case characteristics), assuming that the user will apply explainability to the assessment while looking at these attributes during runtime.

7.4 Discussion

We saw the demonstration of Execution Mode where the dashboard was able to produce the recommendations which was played out on the test log for the case-id “WFA”. Even though it was one of the most frequent flows in the sepsis event log and the dashboard is capable of recommending the actions, it is only limited by the LSTM model recommendation capabilities for other non-frequent traces to be recommended as next action. In the subsequent section, we saw the demonstration of Evaluation mode, which showed that letting the model recommend the next action in a generative way might not be ideal in non-frequent cases. The dashboard is capable of performing conformance check on the pre-select prefix, which helps the user to understand how much the model knows. Lastly, in the what-if mode, the dashboard allows the user to make selection from the recommendations shown on the dashboard and is capable of replicating traces that are similar in terms of contextual information and thus, it can be applied on an ongoing case to perform AI actions directly from the dashboard.

Using the Nunes et al. [10] explainable AI objectives system was able to achieve *effectiveness* and *efficiency* which determines users to make reasonable and fast decisions could be achieved by Execution mode, *education* and *debugging* which allows the users to learn and detect the defects in the system is achieved using Evaluation mode and *scrutability* which enables the user to communicate with the system is achieved using What-If mode.

7.5 Summary

This chapter discussed the demonstration of single event processing modes from a user perspective. Throughout the demonstration, the business user's utilization of the modes was exhibited. The application of One-Step Ahead prediction was demonstrated to assist the user in making decisions at runtime. The Evaluation mode discussed how the user could assess the model quality and the risks associated with it. Apart from this, the What-if mode demonstration showed how our dashboard is capable of communicating with the model as well as performing AI actions on a running trace.

Chapter 8

Conclusion

This thesis introduced a framework for the predictive process monitoring dashboard for business users, specifically the domain experts, which could explain the prediction outcome and provide an alternative recommendation. The framework is designed to incorporate any type of event logs, but in our work, we modelled it around the Sepsis Cases event log. It is a real-life event log that recorded the trajectories of patients with life-threatening sepsis symptoms at a Dutch hospital. Each case records the events that were executed for one patient, from the time the patient checked in at the registration to the time they were discharged. Since this work is a first-of-its-kind, we do not have any state-of-the-art with which it can be compared. We tried to lay down the first academic prototype, and it has its flaws and imperfections.

We first developed the training architecture, which helped in improving the existing sequential deep learning architecture of [6]. We used the random forest classifier for feature selection from the event log to select features that could better explain the activity, called intra-case features. We also included inter-case features such as the number of open cases because, based on recent studies, it improves the accuracy [23, 24]. Along with this, we also included meaningful features extracted from timestamp, such as weekday and hour of the day. The selected features were encoded in a pre-processing phase (n-gram, scaling), and then the LSTM model was trained. While training, we found that encoding all features were leading to overfitting, so we also trained a model which does not overfit. This led to us having three models to compare, the base model and two improved models. Then, using the existing Damerau-Levenshtein [29] similarity measure, we calculated the similarity of the activity and roles and also calculated the accuracy and measured mean absolute error for time duration. The non-overfitting improved model performed better than the base model in all aspects except the similarity measurement of role. In the end, we performed a qualitative evaluation to understand how much the model deviated from the true behaviour. We found that the LSTM model did not learn all the end behaviour of the event, mostly because there were comparatively fewer cases in the event log however, the overfitting improved model was able to learn the most reoccurring paths better than the other models. Although in the conformance check, the non-overfitting model performed better. However, we performed the evaluation and demonstration using the overfitting model with the assumption that it might explain the recommendations better.

Next, we discussed how the chosen predictive model can perform multi-prediction using the softmax function and how we handled the number of recommendations the model could output. Then we outlined the functional requirements of the user interface of the dashboard based on the use case of business users who are domain experts [9]. The conceptual design is intended towards the explainability of the predictive model from a domain expert perspective. After the conceptualization, Streamlit framework was used to build the dashboard. We introduced case by case processing called Single Event Processing, where three features were designed towards replaying of the test event log, of which two of them are also capable of processing running case. These features were Execution Mode, Evaluation Mode and What-If Mode. Each of them is intended towards different applications to help the business user to make decisions. To evaluate these different modes, we also developed Batch Processing which offered to execute the test event log which was to be executed for different predictive techniques used in Single Event processing.

Execution mode aimed to provide different recommendations to the user ordered based on the confidence of the model. It also provides the one step ahead future insight using two different predictive techniques to build confidence in the user before selecting the recommendations. The label assigned by the user for each of the recommendations was employed in this mode to detect if the trace is deviant or regular. Apart from this, contextual information which we had extracted during feature engineering is also displayed, along with the role and time duration. Evaluation mode aimed to provide insight to the user about the model's capabilities and build trust in the predictive model. It is only executed over historical trace (test log), which makes this mode capable of selecting pre-select prefix which can be changed back and forth. This mode shows how the model could have led the trace until the end of the case if the actions were solely selected using its recommendations versus how it would have acted if it was replayed over historical traces. It also offers the conformance checking over the pre-select prefix for the user to understand how much the model knows beforehand. What-If mode allows the user to choose from the variety of recommendations displayed on the dashboard and simulate it by selecting various recommendations offered on the dashboard to reach a successful outcome. This mode is also capable of executing over the running case.

The predictive techniques applied in the Single event processing were offered in different options, namely SME, Generative and Prediction. These were evaluated in the Batch Processing of the dashboard, which executes the entire test event log and compares different recommendations. The evaluation was performed based on the similarity measurement on event level as well as on log level. The event level similarity measurement employed Damerau-Levenshtein distance for activity and role, whereas Mean Absolute Error was used for time. The log level similarity measurement used Control-Flow Log Similarity (CFLS) and Event Log Similarity (ELS). Through our evaluation, we found that the predictive technique used for the SME option works best for providing predictive recommendations. In contrast, the other predictive techniques should only be used to foresee future possibilities in multi-prediction options as they generate similar kinds of recommendations owing to the predictions being stuck in higher probabilities.

Lastly, we looked at the demonstration of different modes from the business user perspective. We emulated different cases for each mode and explained what is seen on the dashboard. In the Execution mode demonstration, the objective was to show corresponding recommendations which the user has selected in the test log for each event for the given case. The case-id we selected successfully demonstrated the objective, although it is not the same with all the case-id we performed experiments on, as there were not enough cases in the event log for the LSTM to recognize them. Next, we demonstrated how the model would have led the case if the recommendations from the model had been followed. The case-id we conducted the demonstration on showed that only adhering to the recommendations by the model would not have generated a successful outcome. We then demonstrated the What-If mode, where we tried to steer two relatively similar cases based on contextual information to follow each other's actions. We were successful in achieving it, but it required the knowledge of concurrency and interchangeability of actions therefore, we first found these patterns by looking into the process model and incorporated them during our experiment to reach the objective of the demonstration.

Now looking at the Márquez et al. [21] research claim that existing PPM tools are not usable because of their complexity and incapability to explain the recommendation. The dashboard we developed solves this, and it is the first of its kind, built from scratch only intended for real-world business users who understand the process and are considered domain experts. The framework is the starting point that could be enhanced in future work. Also, we tested the model prediction by demonstrating from a user perspective and reported how the dashboard performs. The objective we set was achieved for each demonstration. Also, we included the contextual information, which the paper suggested other PPM models lagged. Finally, we were able to achieve most of the explainable AI objectives for the business user listed by Nunes et al. [10] using the features of Single Event processing, which covered effectiveness, efficiency, education, debugging and scrutability. Out of all the objectives' domain experts only require effectiveness according to [21]. However, based on Mehdiyev et al. [9] domain expert's use case, our dashboard is capable of building trust, case by case explanation and able to justify the model outcome.

8.1 Limitations and Future Work

The first and foremost major limitation is the predictive model. The overfitting during training led it not to incorporate all the traces which were there in the training event log. Thus, the framework demands a more accurate deep learning-based predictive model design.

Next, in the evaluation among the multi-recommendations, it became apparent that the recommendations from the predictive model could not be completely relied on as the business acumen is essential for handling the situation. Since we lacked knowledge of the Sepsis Process, we tried to form assumptions based on what we saw using the process model. So, the future work can include the event log for which business knowledge is documented, or the framework could be handed to the business user to be tested by them and further conduct the survey what the dashboard would have shown to let them decide on the next actions.

The labelling of recommendations was conducted at the prediction end, which made it user-customizable; however, business processes have predefined metrics as KPI's using which a supervised model can be trained and integrated on the predictive side to predict if the recommended actions are going to be deviant or regular.

The predictive techniques employed are not optimized for fast execution, which could be improved by using a better algorithm and parallel processing, as it is quite slow from the dashboard standpoint.

The temporal and resource capability is untapped in our evaluation, while explanation could be handled in future work.

The dashboard designed could only handle one case at a time, but it is not the case in the real-world event stream. Our dashboard is incapable of handling multiple cases at once by the same user. Although multiple users can use the dashboard at the same time and work on different cases, but it's not ideal. Thus, the design needs a mechanism to save the case state and execute on them as the case progresses until the end of the case is achieved.

What-if mode predictive recommendations for activity and role are grouped together based on the confidence, and this is not ideal considering modularity, the framework could be enhanced for selecting activity and role individually before generating the next predictive recommendation.

The dashboard only included one predictive model recommendation. However, the multimodel recommendation might be more apt for this situation because then the user with limited business knowledge could also use the dashboard, where the decision would be taken from the majority point of view.

Bibliography

[1] W. van der Aalst, “Data Science in Action,” in Process Mining, pp. 3–23, Springer Berlin Heidelberg, 2016.

[2] F. M. Maggi, C. D. Francescomarino, M. Dumas, and C. Ghidini, “Predictive Monitoring of Business Processes,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8484 LNCS, pp. 457–472, 2014.

[3] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, “Fundamentals of business process management: Second Edition,” Fundamentals of Business Process Management: Second Edition, pp. 1–527, 3 2018.

[4] N. Tax, I. Verenich, M. La Rosa, and M. Dumas, “Predictive business process monitoring with LSTM neural networks,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10253 LNCS, pp. 477–492, Springer, Cham, 2017.

[5] J. Evermann, J. R. Rehse, and P. Fettke, “Predicting process behaviour using deep learning,” Decision Support Systems, vol. 100, pp. 129–140, 8 2017.

[6] M. Camargo, M. Dumas, and O. González-Rojas, “Learning Accurate LSTM Models of Business Processes,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11675 LNCS, pp. 286–302, Springer, Cham, 9 2019.

[7] E. Rama-Maneiro, J. C. Vidal, and M. Lama, “Deep Learning for Predictive Business Process Monitoring: Review and Benchmark,” 9 2020.

[8] I. Teinemaa, M. Dumas, M. L. Rosa, and F. M. Maggi, “Outcome-oriented predictive process monitoring: Review and benchmark,” 3 2019.

[9] N. Mehdiyev and P. Fettke, “Explainable Artificial Intelligence for Process Mining: A General Overview and Application of a Novel Local Explanation Approach for Predictive Process Monitoring,” in Studies in Computational Intelligence, vol. 937, pp. 1–28, Springer, Cham, 2021.

[10] I. Nunes and D. Jannach, “A systematic review and taxonomy of explanations in decision support and recommender systems,” User Modeling and User-Adapted Interaction 2017 27:3, vol. 27, pp. 393–444, 10 2017.

[11] W. Van Der Aalst, A. Adriansyah, A. K. A. De Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. Van Den Brand, R. Brandtjen, J. Buijs, A. Burattin, J. Carmona, M. Castellanos, J. Claes, J. Cook, N. Costantini, F. Curbera, E. Damiani, M. De Leoni, P. Delias, B. F. Van Dongen, M. Dumas, S. Dustdar, D. Fahland, D. R. Ferreira, W. Gaaloul, F. Van Geffen, S. Goel, C. Günther, A. Guzzo, P. Harmon, A. Ter Hofstede, J. Hoogland, J. E. Ingvaldsen, K. Kato, R. Kuhn, A. Kumar, M. La Rosa, F. Maggi, D. Malerba, R. S. Mans, A. Manuel, M. McCreesh, P. Mello, J. Mendling, M. Montali, H. R. Motahari-Nezhad, M. Zur Muehlen, J. Munoz-Gama, L. Pontieri, J. Ribeiro, A. Rozinat, H. Seguel Pérez, R. Seguel Pérez, M. Sepúlveda, J. Sinur, P. Soffer, M. Song, A. Sperduti, G. Stilo, C. Stoel, K. Swenson, M. Talamo, W. Tan, C. Turner, J. Vanthienen, G. Varvaressos, E. Verbeek, M. Verdonk, R. Vigo, J. Wang, B. Weber, M. Weidlich, T. Weijters, L. Wen, M. Westergaard, and M. Wynn, “Process mining manifesto,” in Lecture Notes in Business Information Processing, vol. 99 LNBIP, pp. 169–194, Springer, Berlin, Heidelberg, 2012.

[12] W. M. P. van der Aalst, Process Mining. Springer Berlin Heidelberg, 2011.

[13] I. Verenich, M. Dumas, M. La Rosa, F. M. Maggi, and I. Teinemaa, “Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring,” ACM Transactions on Intelligent Systems and Technology, vol. 10, 7 2019.

[14] J. Schmidhuber, “Deep Learning in neural networks: An overview,” 1 2015.

[15] X. Hao, G. Zhang, and S. Ma, “Deep Learning,” in International Journal of Semantic Computing, vol. 10, pp. 417–439, World Scientific Publishing Company, 11 2016.

[16] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” 5 2015.

[17] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, pp. 1735–1780, 11 1997.

[18] C. Di Francescomarino, C. Ghidini, F. M. Maggi, and F. Milani, “Predictive Process Monitoring Methods: Which One Suits Me Best?,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11080 LNCS, pp. 462–479, 9 2018.

[19] C. Guo and F. Berkhahn, “Entity Embeddings of Categorical Variables,” 4 2016.

[20] L. Lin, L. Wen, and J. Wang, “MM-Pred: A deep predictive model for multi-attribute event sequence,” in SIAM International Conference on Data Mining, SDM 2019, pp. 118–126, Society for Industrial and Applied Mathematics Publications, 2019.

[21] A. E. Marquez-Chamorro, M. Resinas, and A. Ruiz-Cortes, “Predictive monitoring of business processes: A survey,” IEEE Transactions on Services Computing, vol. 11, pp. 962–977, 11 2018.

[22] M. Camargo, M. Dumas, and O. González-Rojas, “Discovering Generative Models From Event Logs: Data-driven Simulation Vs Deep Learning,” PeerJ Computer Science, vol. 7, pp. 1–23, 7 2021.

[23] R. Conforti, M. De Leoni, M. La Rosa, W. M. Van Der Aalst, and A. H. Ter Hofstede, “A recommendation system for predicting risks across multiple business process instances,” Decision Support Systems, vol. 69, pp. 1–19, 1 2015.

[24] A. Senderovich, C. Di Francescomarino, C. Ghidini, K. Jorbina, and F. M. Maggi, “Intra and Inter-case Features in Predictive Process Monitoring: A Tale of Two Dimensions,” pp. 306–323, 2017.

[25] G. Tello, G. Gianini, R. Mizouni, and E. Damiani, “Machine Learning-Based Framework for Log-Lifting in Business Process Mining Applications,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11675 LNCS, pp. 232–249, 9 2019.

[26] M. Song and W. M. van der Aalst, “Towards comprehensive support for organizational mining,” Decision Support Systems, vol. 46, pp. 300–317, 12 2008.

[27] Micci-BarrecaDaniele, “A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems,” ACM SIGKDD Explorations Newsletter, vol. 3, pp. 27–32, 7 2001.

[28] F. Taymouri, M. L. Rosa, S. Erfani, Z. D. Bozorgi, and I. Verenich, “Predictive business process monitoring via generative adversarial nets: The case of next event prediction,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12168 LNCS, pp. 237–256, Springer, Cham, 9 2020.

[29] F. J. Damerau, “A technique for computer detection and correction of spelling errors,” Communications of the ACM, vol. 7, pp. 171–176, 3 1964.

[30] J. C. A. M. Buijs, B. F. Van Dongen, and W. M. P. Van Der Aalst, “On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery,”

[31] M. Camargo, M. Dumas, and O. González-Rojas, “Automated discovery of business process simulation models from event logs,” Decision Support Systems, vol. 134, p. 113284, 7 2020.

[32] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” 11 2018.

[33] T. M. H. Reenskaug, “The original MVC reports. Technical report, Xerox PARC.,” 1979.

[34] G. E. Krasner and S. T. Pope, “A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System,” Journal Of Object Oriented Programming, vol. 1, no. 3, pp. 26–49, 1988.

[35] M. Dumas, “Constructing Digital Twins for Accurate and Reliable What-If Business Process Analysis,” 2021.

[36] H. W. Kuhn, “The Hungarian method for the assignment problem,” Naval Research Logistics Quarterly, vol. 2, pp. 83–97, 3 1955.

List of Figures

List of Tables

Appendix A

Process Model

A.1 Sepsis Cases Concurrent Behaviour

Figure A.2 shows the number of relatively less frequent behaviour when the IV Antibiotic is followed by IV Liquid.

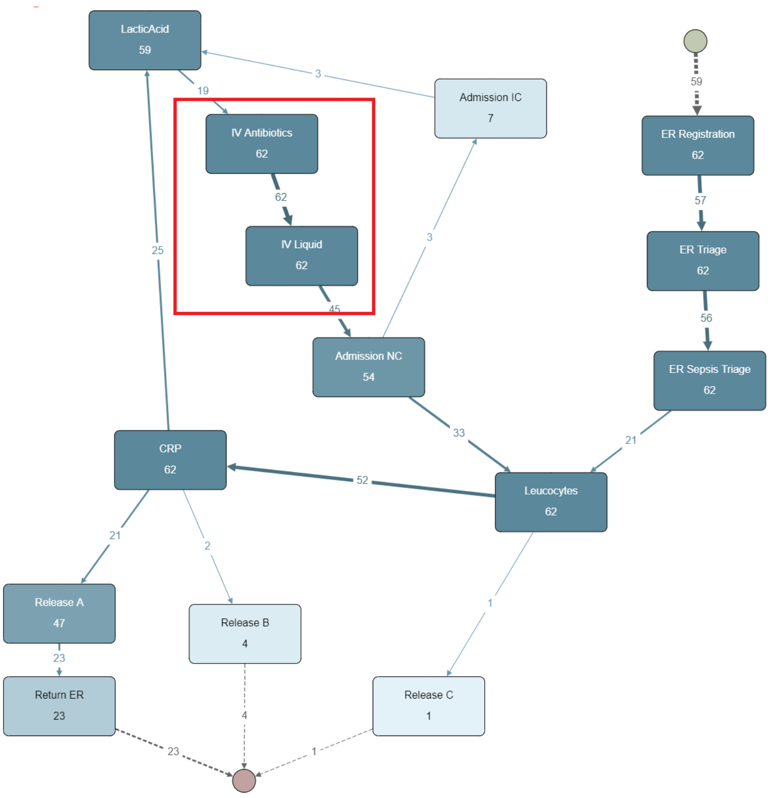


Figure A.1: Sepsis Case IV Antibiotic followed by IV Liquid

Figure A.2 shows the number of very less frequent behaviour when the ER Sepsis Triage is followed by ER Triage.

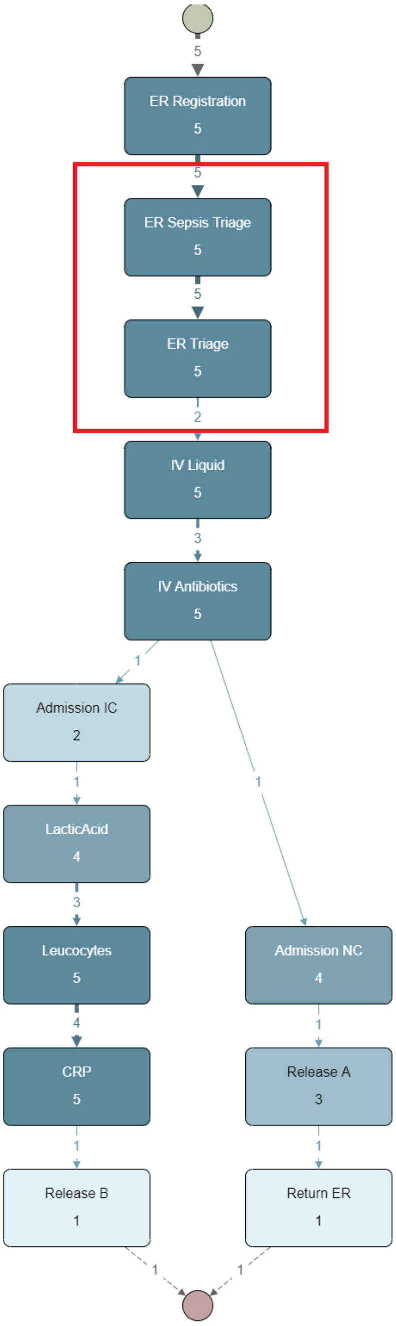


Figure A.2: Sepsis Case Severity Activities when ER Sepsis Triage is followed by ER Triage

Figure A.3 shows the concurrency among the Leucocytes, LacticAcid and CRP, also referred as set of diagnostic activity at 30% Arcs.

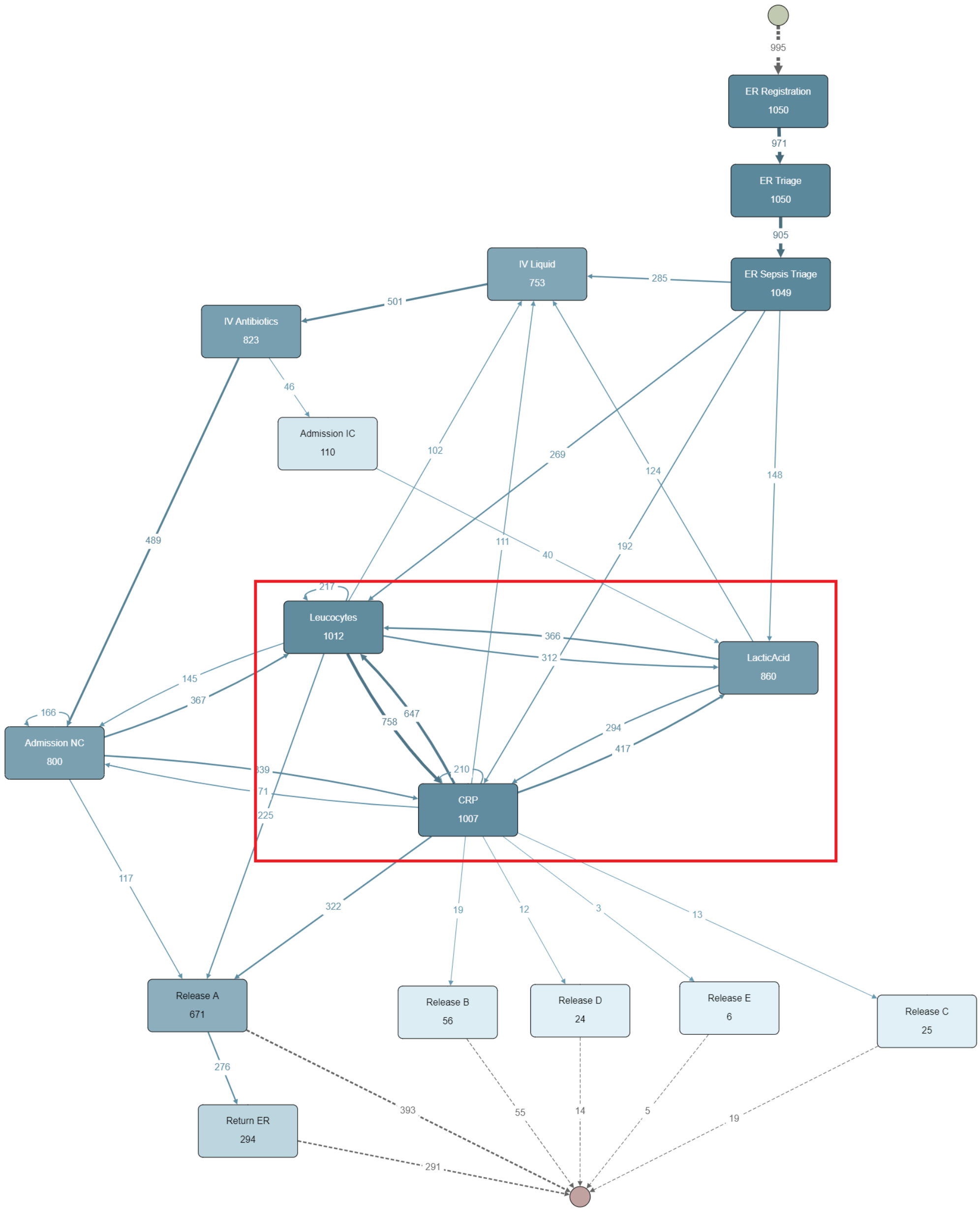


Figure A.3: Sepsis Case Diagnostic Activities Concurrency at 30% arc