BUSINESS PROCESS OUTCOME PREDICTION USING ATTENTION-BASED LSTM

[1]Thuzar Hnin

Faculty of Information Science,
University of Computer Studies, Yangon,
Myanmar,
[1]thuzarhninn@ucsy.edu.mm, [1]thuzarhninn99@gmail.com,

[2]Tin Thein Thwel

Faculty of Information Science,
University of Computer Studies, Yangon,
Myanmar,
tintheinthwel@ucsy.edu.mm

Abstract

Today's many contemporary businesses have already adopted business process management in order to gain competitive advantages predictive business process monitoring and ongoing procedure in business optimization, in particular, are still difficult for businesses to implement as part of a wider procedure in business management program. For the purpose of predicting the end of an ongoing case as early as feasible, predictive business process monitoring examines events generated throughout the performance of a business procedure. Several methods exist for predicting the results of business procedures have been put forth in the literature. However, research on deep learning neural networks, particularly long short-term memory, has just lately been done for monitoring prediction processes. Moreover, the experimental analysis on real-life event logs and the benchmark datasets have also been done. This research's findings show that a long short-term memory with attention provides high accuracy in predictive power than LSTM without attention.

Keywords: **Attention, Deep Learning, LSTM, Predictive Process Monitoring.**

1. Introduction

The predicting of upcoming data from a business process in execution is known as business process monitoring. It is one of the branches of process mining, a method for discovering, observing, and improving actual business processes via information extraction from event logs [3-6]. It allows us to know what actually going on in a business procedure, and not what it thinks is happening [1]. Predictive monitoring approaches are proactive movement before a harm happens to enhance business process performance and to transfer risks. It could include predicting the subsequent event or series of events in a running case, the amount of time left in the case, or the potential resolution of a running case. For instance, it might allow predicting the traffic road fines are either paid in full or not in traffic road fines management system.

Many techniques have been applied to perform these predictions in PPM. To make prediction, while some techniques explicitly depend on the representations of process model, for example a probabilistic finite automaton, or a petri net, and use the extracted feature vectors to train machine learning models such as clustering analysis support vector machines, in recent, deep learning-based techniques [17, 7]. As far as these have learned that the latter have obtained the best results.

Today, DL models have been widely employed in the business processes for predictive monitoring. Recurrent neural networks (RNN), which are experts in sequence processing and were well suited to handle the predictive monitoring problem considering that business operations are sequential [12]. However, there are some issues in evaluating predictive monitoring approaches. They are the great number of combinations of possible architectures, datasets which are the high inequality and the usage of experimental setups.

The contributions of the paper are that propose LSTM with attention mechanism for PPM, in which it applied self-attention layer self-attention layer and one hot encoding for feature extraction. Additionally, it used the DL4J framework to create this suggested network. This open-source framework contains the first distributed deep-learning library that is of commercial quality and written in Java and Scala. After that, this will make

evaluating on three event logs from real-life to prove that the attention-based LSTMs gives it the great results in predicting of process models.

2. BACKGROUND OF APPLICATION DOMAIN

This section provides predictive process monitoring in brief and its related definitions and basic terms by following the standard notations of paper [1].

2.1 Predictive (Business) Process Monitoring-PBPM

Predictive process monitoring's objective, a subfield of process mining, is to predict future behaviors from execution (business) process occurrences. Predictive process monitoring often involves two stages: offline training of a prediction model using historical data, and online prediction using data from active processes. Case prefixes are retrieved from an event log, filtered, and categorized into homogeneous buckets of "similar" prefixes. In the offline phase, prefixes are then encoded into feature vectors from each bucket. After being encoded, these feature vectors are used to fit a machine learning model. The actual forecasts for ongoing instances are created in the online phase by recycling the offline phase's buckets, encoders, and predictive models [4,8,14-16]. As far as is known, zero bucketing, prefix length bucketing, cluster bucketing, and state bucketing are the four most often utilized bucketing techniques. Other prefix encoding methods include [6] the ones listed below:

1) Aggregation encoding
2) Last state encoding
3) Index-based encoding
4) Tensor (LSTM) encoding

The following categories are used to categorize predictive business process monitoring depending on the sort of prediction being made [14,15]:

1) Prediction of remaining time, it is also known as regression tasks,
2) Prediction of next activity, it is also known as multi-class classification, and
3) Prediction of business process's outcome, it is also known as binary classification.

Among the different prediction results, while business process is executing, though remaining time prediction is very important, the focus of this paper is the outcome (normal or deviate) oriented prediction for business process behaviors by also considering other process metrics. For a business process, it is certain that the event logs produced during the process's execution can serve as a valuable source to predict with different results and various purposes, including resource allocation, risk management by foreseeing compliance violations, assessment of technical production parameters for more effective production plan optimization, analysis of the customer behavioral patterns, and others. In certain, it exploits event logs, which are more and more widespread in modern information systems, to enable users to forecast the course of ongoing (unfinished) cases up until completion. Usually, an event log is made of events, which provide a case identifier, an activity, and a timestamp without a doubt. But there are also event attributes, which are unique to each event, and case attributes are values that are common to all of the events in a certain case are included in option [1]. Table 1 is the example of event log [1].

Case Id	Event Id	Properties			
		Timestamp	Activity	Resource	Cost
C1	E01	30-12-2010:11.32	Register request	Mike	50
C1	E02	30-12-2010:12.12	Check ticket	Mike	100
C1	E03	30-12-2010:14.16	Examine casually	Pete	400
C1	E04	05-01-2011:11.22	Decide	Sara	200
C1	E05	08-01-2011:12.05	Pay Compensation	Ellen	200
C2	E06	30-12-2010:14.32	Register request	Pete	50
C2	E07	30-12-2010:15.06	Examine casually	Mike	100
C2	E08	30-12-2010:16.34	Check ticket	Ellen	50

Table 1 Example of Evet Log.

2.2 Definitions and Terms of BPM

To be clearer, these will refer to some definitions of paper [1]. The paper [1] has described six basic definitions of process monitoring and ten definitions which are related to predictive monitoring they will reinterpret a few definitions, most of which are relevant to this paper's work, outcome-oriented prediction of business process.

Definition 1. Assume that A represents the range of possible activities, C represents cases in the universe, T represents the time domain, and D1, ..., Dm reflects the realms of each attribute of the log's traces and events, with m ≥ 0. An event e ϵ E is a tuple (a; c; t; d1, ..., dm) where a ϵ A, c ϵ C, t ϵ T and di ϵ {Di U e} with i ϵ [1; m] and e being the empty element.

Each event in a process log is distinct, meaning that no two events can occur simultaneously in the same situation with the same action.

Definition 2: Assume that πA, πC, πT and πDi are the functions that translate an event into an activity, a case identifier, a timestamp, and an attribute, respectively, then,

$$\pi A(e) = a \quad (1) \qquad \pi C(e) = c \quad (2) \qquad \pi T(e) = t \quad (3) \qquad \pi Di(e) = di \quad (4)$$

For the events ei and ej, since each event is different, they can be described as:

$$ei; ej \in E : ei \neq ej \rightarrow \pi A(ei) \neq \pi A(ej) - \pi C(ei) \neq \pi C(ej) - \pi T(ei) \neq \pi T(ej). \quad (5)$$

For example, the second event of Table 1 is E02, with πA(e2) = "check ticket", πC(e2) = "C1", πT (e2) = "30-12-2010 12.12" and πD1(e2) = "Mike". A trace is a list of incidents that all have the same case identification. Each event in this series has a timestamp that is equal to or greater than its forerunner.

Definition 3: Assume that S is the universe of traces. a trace σ ϵ S is a non-empty sequence of events σ = <e1, …, en> which holds that ∀ ei; ej ϵ σ; I, j ϵ [1; n]: j > i ^ πC(ei) =πC(ej) ^ πT (ej) ≥ πT (ei) where |σ| = n. As an illustration, the first trace in Table 1 has 4 events from the case "C2." In addition, a set of traces can be used to define an event log.

Definition 4. A set of traces, L = {σ1, ..., σl} such as L = {σi | σi ϵ S ^ i ϵ [1; l]} where |L |= l, constitute an event log. The log seen in Table 1 is made up of two traces that are connected to the cases "C1" and "C2," continuing with the Evet Log example. Each log trace will inevitably have a specific conclusion ascribed to it.

The following definitions are linked to the forecasting of business process outcomes which will be described in the definition 6 and definition 10. They are recharacterized as follows because it also emphasizes outcome-oriented predictive process monitoring:

Definition 5. A domain dependent label that delivers details about the entire case of interest is the result of a running case. Let hdk be the k-length event prefix. Let 'O' be the universe of outcomes, where o ϵ O. Then, πO is a function that maps an event prefix to an outcome such as πO(σ) = πO(hdk(σ)) = o. Possible outcomes include things like whether a case will be reopened in the future or if a customer's order will be accepted.

Definition 6. The outcome of an event prefix can be predicted as ΩO hdk(σ)) = πO(hdk(σ)) = o. let hdk(σ) be an event prefix such as hdk(σ)=⟨e1, ek⟩, e′ be a predicted event by a function Ω. In here, the function of Ω will be the function of LSTM neural network.

In the literature, several approaches have been proposed to deal with a typical predictive monitoring problem. Two key articles by [8,9] are kindly cited for the detailed systematic literature review (SLR). Motivated from these readings, these have also briefly done literature review, which are only focusing on deep learning approaches applied in PBPM [21] as this paper's previous work.

3. RELATED WORKS OF LSTM

There is still opportunity for research even if business process prediction has received attention with various goals, many data sources, and a variety of different approaches like decision trees, the support vector machine (SVR) approach, naïve Bayes (NB), and clustering techniques. Moreover, some researchers have worked on an explicit process model, e.g., mined from event logs, while others have worked on an implicit process model (neural networks) to predict after introduced by [11] of DL to apply in this field. Due to the sequential character of process traces, the paper [11] has noted that predictive process monitoring corresponds to NLP. As a result, the motivation obtained from the successful achievement of DL to NLP is a natural fit to the problem of process prediction. And then the various prediction models, CNN, LSTM and RNNs have been suggested for the tasks of next activity [17-19], creating suffix [18,14], predicting outcomes [11], and prediction of process's remaining time.

The authors of the research [10] describe how PPM uses LSTM neural networks to forecast the next step in a running business process scenario. Also, the author has talked about how LSTM networks are used to forecast outcomes. After that, the author released a second study [10] in which they defined the idea of temporal stability for monitoring predictive processes and compared their suggested method to current ones in terms of precision

and consistency of time. The studies were conducted using 6 real-world, publicly accessible datasets and 12 prediction tasks. The authors also discovered that only one classifier strategy using XGBoost is used first, then LSTM, yields the greatest temporal stability. Recurrent neural networks (RNNs) are used to forecast outcome-focused monitoring of the prediction process is new, at least as far as this research is aware. Long short-term memory units (LSTMs) in RNNs have been utilized for prediction of remaining time and the next task [10,11]. Although deep RNNs-based PBPM techniques have been well developed for modeling event sequences, they have significant drawbacks, particularly when it comes to the difficulty of anticipating the upcoming action [2].

In order to obtain the numerical representations of categorical event sequences, several proposed approaches use one-hot encoding, as the authors have first noted. Due to an increase in data dimensionality, these numerical representations overlook the underlying relationships between events and impose prohibitive processing requirements. Second, the research [2] addressed how a clear separation between long-range and short-range modeling relationships is necessary for LSTM to avoid performance degradation because the length of event sequences directly affects performance. However, it is very undesirable for event logs due to links made by control flows across activities. Lastly, parallelization is impossible because to the LSTM and RNNs' inherent sequential character, which leads to severely ineffective inference and learning. Without taking into account their proximity in the input and output sequence, the attention mechanism is proposed as a solution to the long-range dependency problems for sequence modelling [22].

[22] also proposed a deep sequence model called the transformer neural network architecture that leverages self-attention to keep extended sequences coherent. These models have quickly taken over as the main architecture for neural machine translation and NLU [23]. In particular, the revolutionary language understanding models generative pretrained transformer (GPT-3) and bidirectional encoder representations from transformers (BERT) are based on the transformer architecture. Transformers haven't been investigated regarding business process management despite their impressive performance in numerous sequence modeling challenges. Thus, the purpose of this work's contribution is to another potential of LSTM using attention mechanism by adopting the knowledge from [10,11], for outcome oriented predictive process monitoring.

4. PROPOSED NEURAL NETWORK MODEL

This section describes the proposed neural network model using self-attention-based LSTM for outcome-oriented prediction of business process. In the previous research work, it is studied about a brief review of deep learning methods for monitoring prediction process published in [20]. Then, among the various DL network models, these have also studied and made systematic literature review (SLR) about LSTM and GRU network models in [20,21]. To apply DL in this motivation obtained, an implicit process model (neural networks) is based on prediction.

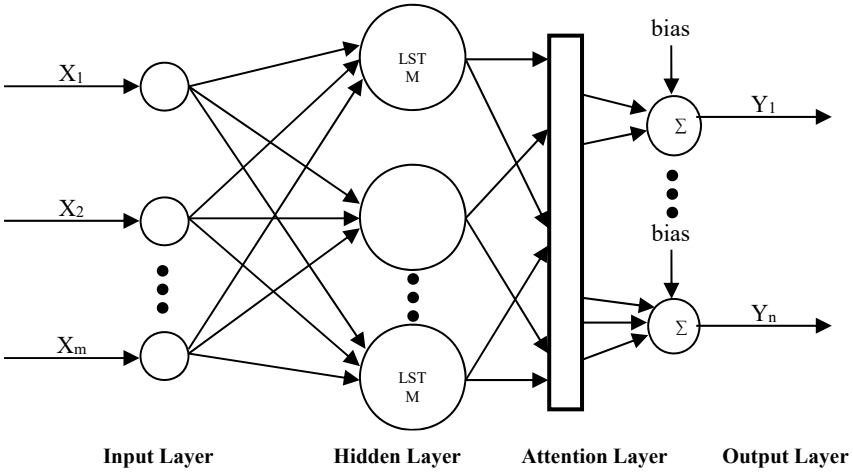

Figure 1 Proposed System Model

The paper pointed out that predictive process monitoring is conforming to the NLP because of the sequential nature of process traces. As a result, the motivation getting from the successful achievement of DL to NLP is a natural fit to the problem of process prediction. Now, some issues of LSTM and then how to make pre-processing, feature encoding of the implementation will be discussed in following sub sections.

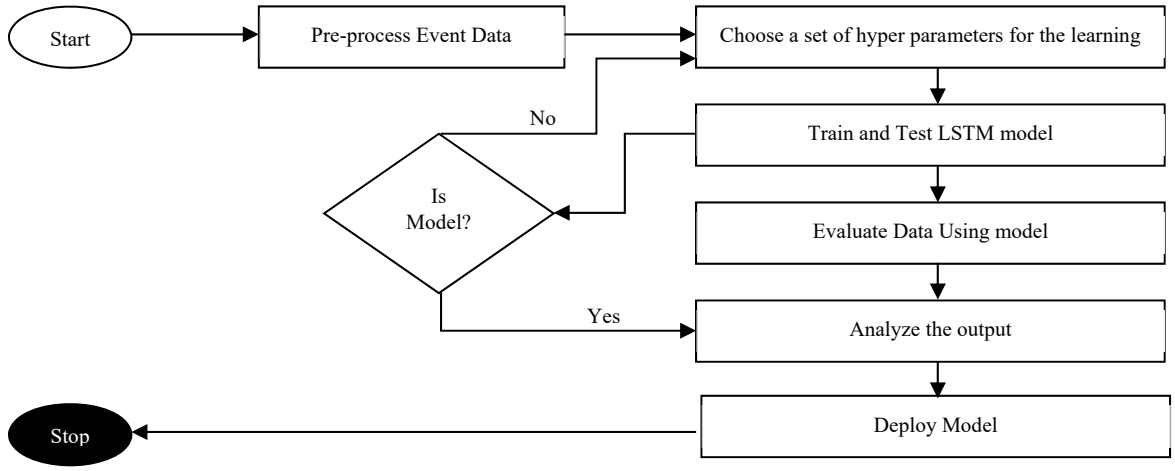

Figure 2 System Flowchart for Implementing the Proposed Model

4.1 Issues of LSTM

The ordinary LSTM can deliver consistently high accuracy in prediction of business process and remind the previous information for future prediction. However, some issues of LSTM can be found:

It may not be able to identify significant affects from older timestamps because it is limited to only a few most recent steps, which have more impact than previous ones.

A fixed length internal representation is created by encoding the input sequence. As a result, there are restrictions placed on how long input sequences can be learned rationally, and performance is negatively impacted for extremely long input sequences.

As it relates to how they perform declines according to the length of event sequences, long- and short-range relationships are not explicitly modelled in LSTM. The interconnections that control flows among activities introduce into event logs make it particularly undesirable.

Finally, because LSTM and RNNs are inherently sequential, parallelization is impossible, which leads to drastically inefficient learning and inference.

Although the standard LSTM can continuously give high accuracy in business process prediction and recall the past information for future prediction, as was described in section 3, it cannot. It cannot identify significant affects from previous timestamps since it can only analyze the most recent steps, which have a greater impact than earlier ones. It transforms the incoming input sequence into an internal representation with a defined length. As a result, there are limitations on the length of input sequences that can be learned well. Moreover, this has a negative impact on performance for excessively long input sequences. To address these problems, attention mechanism is needed in recent trend for deep learning. Moreover, according to the literature review, for PPM, LSTM models has limitation when dealing with traces that contain several instances of a business process or activity, or when the model anticipates excessively long sequences of the same event. As the result, this research work aims to propose to use self-attention-based LSTM architecture for monitoring of prediction processes to address the issues of LSTM. The proposed design with an attention mechanism is shown in Figure 1.

4.2 Attention Mechanism

Though the attention mechanism is originated form human vision system, in [20]. [20] have introduced the attention mechanism to solve the bottleneck issue caused by a fixed length encoding vector, which would restrict the decoder's ability to access the input's information. This is thought to be especially problematic for long and/or complex sequences as the representation's dimensionality would be limited to match that of shorter or simpler sequences. This method, which is frequently used with RNN, LSTM, BiLSTM, and other techniques, tries to provide more weight to the information that is most pertinent to the input data rather than all the information available [24].

For several different deep learning models in numerous domains and tasks, attention is a crucial mechanism that can be used [25]. An overview of the most significant attention mechanisms mentioned in the literature is given in this work. Using a framework made up of a broad attention model, consistent notation, and an extensive taxonomy of attention mechanisms, the numerous attention mechanisms were discussed. A model can employ various combinations of the several sorts of attention processes and extensions that are available.

Due to this, a taxonomy that can be used to categorize various attentional mechanisms has been developed in [25]. To be more clear information, the readers are referred to the papers [25], [13], [24] and [26].
Based on the knowledge from the above-mentioned reference papers and to address the issues of LSTM, so as to forecast the results of business procedures, the system in this research suggests using self-attention-based LSTM. One of the most important kinds of attention is likely self-attention in part because of its crucial significance in the enormously well-liked transformer model. Nonetheless, it is a fairly universal mechanism that may be used to solve almost any issue. As a result, self-attention has been thoroughly investigated in a variety of domains using models and architectures based on transformers, which are explored for image recognition tasks, generative adversarial network (GAN), medical image segmentation model, for video processing and video summarization, a speech recognition model, next item recommendation, audio processing, sentiment analysis and so on. Due to self-attention reaching outcomes from cutting-edge research in several fields, it became the motivation to apply Self-attention mechanism after LSTM layers in the proposed architecture. Figure 2 displays the system flow of implementing the proposed DL model. Firstly, these will choose the datasets to use from the benchmark dataset and do pre-processing of event data by using feature encoding. Then, these will make initial parameter settings for learning algorithm and tune hyperparameters for optimal results. After that, the training and testing of the proposed LSTM model before doing comparative analysis is presented in this paper.

4.3 Feature Encoding

As everybody knows, a good representation and understanding of data can make more accuracy in prediction analysis. Though regression and classification analysis can use categorical variables, deep learning algorithms accept only numeric values as input. Therefore, it is inevitability needed to do feature encoding as pre-processing step. In the literature, there are a lot of encoding techniques to feed data to the DL model. Among them, this paper uses One-hot encoding technique. Pre-processing categorical features for machine learning models frequently involves using one-hot encoding. It converts a categorical value into a binary value vector, where a value of 1 denotes the occurrence of one element and a value of 0 denotes the absence of the remaining items. This encoding method is the subject of this paper due to its ease of implementation and independence from the data processing and classification models. Figure 3 shows the encoding sample to get feature vector from one record of dataset these used. After processing the data and reshaping it to fit the proposed models, the paper moves on the training and testing steps for this paper's model.

Attributes Name	Record1	Record2	Record3
Speciality	C	I	M
Case ID	RB	R	PB
Activity	NEW	NEW	NEW
Resource	ResN	Other	ResNA
actOrange	Missing	Missing	Missing
actRed	Missing	Missig	Missing
Blocked	FALSE	FALSE	FALSE
caseType	B	A	B
Diagnosis	Missing	S	Missing
flagC	Missing	Missing	Missing
flagD	TRUE	TRUE	FALSE
msgCode	Missing	Missing	Missing
msgType	Missing	Missing	Missing
State	In progress	In progress	In progress
Version	Missing	Missing	Missing
isCancelled	FALSE	FALSE	FALSE
closeCode	Missing	Missing	Missing
msgCount	0	0	0
Complete Timestamp	12/13/2012 11:13	12/13/2012 12:12	12/13/2012 11:18
Timesincemidnight	673	732	738
Month	12	12	12
Weekday	3	3	3
Hour	11	12	12
Timesincelastevent	0	0	0

[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1,
0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0.0, 1355397198000, 673,

Figure 3 One-hot encoding Sample

5. EVALUATION

At first, for this paper's experiments, the chosen datasets will be presented in this part. Then the implementation of the experiments, including splitting the datasets into training and testing, hyperparameter tuning are described. After that the paper will describe the evaluation metrics these used and this paper's results in each subsection.

The evaluation results carried out on the trained attention-based LSTM network model are discussed in this section. Whenever prediction task is performed, "accuracy" is common usage metric used to evaluate models for classification.

The percentage of cases that were correctly predicted out of all instances is what is known as accuracy. For accuracy calculations, the prediction of the model only needs to compare with real classes. From that, the accuracy is calculated using the following formula.

$$\text{Accuracy} = (tp + tn)/(tp + tn + fp + fn) \qquad (6)$$

This equation states that the accuracy increases as the number of accurate predictions increases. Additionally, the accuracy value ranges from 0 (which denotes that there is no valid prediction) to 1 (which denotes that all predictions are accurate). In contrast to recall, which is the actual positive rate for a given class, precision is the number of activities that were correctly classified for a given class given all of its predictions.

5.1 Benchmark Datasets

After the authors [8] have conducted a SLR, they have done a comparative of 11 methods using 9 real-life benchmark datasets including the eight datasets are public datasets, one is a private dataset that is the historical records of the claim handling procedure at Australian insurance company, and both are accessible from the 4TU institute for research data. Among them, they employ three datasets for evaluation, including small and large datasets connected to real-world business:

Production: This dataset describes the event log of manufacturing process. It includes 2490 records with 21 input features and 1 label output feature.

Attributes	**Record 1**	**Record 2**
Part_Desc_	Cable Head	Electrical Contact
Rework	Missing	missing
Work_Order_Qty	250	320
Case ID	Case178	Case238
Activity	Round Grinding - Machine 3	Turning & Milling - Machine 4
Resource	ID4445	ID4794
Report_Type	D	S
Resource.1	Machine 3 - Round Grinding	Machine 4 - Turning & Milling
Qty_Completed	31	0
Qty_for_MRB	0	0
activity_duration	290	360
Complete Timestamp	1/2/2012 4:50	1/2/2012 7:00
timesincemidnight	290	420
month	1	1
weekday	0	0
hour	4	7
timesincelastevent	0	0
timesincecasestart	0	0
event_nr	1	1
open_cases	1	2

Table 2 Sample Data Records from "Production" dataset

Hospital Bill: This data set was retrieved from an area hospital's ERP system's financial modules. Events pertaining to the hospital's invoicing for medical services rendered are noted within the log of events Each trace

of the log of events includes a record of the actions taken to bill a group of medical services that were packaged together. The real medical services that the hospital offers are not disclosed in the incident log. 27 input features and 1 output (label) feature are present in 428627 records.
Traffic Fines: This is the event log of traffic road fining system. Fines are either paid complete (normal) or sent for credit recovery (deviant). It includes 460556 number of records and 21 input features and 1 label output feature.
The number of cases different from 225 in the production log to 129 615 in the traffic fines log, according to the data statistics in [5]. Only roughly 5% of cases in the hospital billing dataset belong to the positive class, making it the dataset with the worst class imbalance. Also, the creation of traffic fines is virtually perfectly distributed among the classes. The files for datasets extension types include:

"csv" (comma separated value) file or
". xes" file (Event Stream extensible).

The Task Force on Process Mining of the IEEE has embraced this, and tools like ProM and Disco are compatible with the format. Instance records of production dataset files are shown in table 2.

5.2 Implementation and Experiment Setup

The proposed model, as represented in Figure 1, has four essential parts:

1) The input layer should contain the model's feature-encoded event log data.
2) Use LSTM layers in the hidden layer to extract better features from the input put layer.
3) Using the self-attention mechanism, the third layer of attention generates a weight vector that equilibrates the hidden states of each phase (steps) and focuses attention on the more significant ones among the sequence of hidden state information.
4) The architecture's fourth layer, known as the output layer, is where sequence-level feature vectors are obtained and used for event log data analysis and prediction.

These implement the above-mentioned architecture model by using java based DL4J framework. The 20% if dataset is separated as test set for the final evaluation of the model's predictions. The proposed model is implemented and run on the machine with Core (TM) Intel (R) i5-3320 MCPU@ 2.66GHz, RAM 8.00 GB, 64GB memory, and Windows 10. In the implementation, for evaluation and comparative analysis, these build four network architectures such as simple ANN, original RNN, original LSTM which means LSTM network without attention mechanism and the proposed self-attention-based LSTM along with the following detailed hyperparameters.

Hyperparameters	**Value Range**
Number of hidden layers	2 ~ 6
Number of neurons	100~300
Learning Rate	0.01~0.9
Batch Size	50~ (total Examples / 1000)
Epoch	50~100
Optimizer/Updater	SGD, Nesterovs

Table 3 Hyperparameters for implemented network model

The gradient descent algorithm's hidden layer count, neurons per layer, batch size, epoch, starting learning rate, and optimization algorithm employed are mainly involved in the configuration of the neural network architecture to get sequence-level feature vectors, which are subsequently used for event log data analysis and prediction. These are shown in Table 3.
After running on the above hyperparameters setting, the JSON file of trained model of this research's proposed model can be obtained and saved for later usage for testing. Indeed, while the model is being trained, the output of the model is 0 or 1. the label, "deviates" or "regular". The example is shown in Definition 5. The use of several hyper parameter settings in a range of different value ranges was demonstrated in the aforementioned Table 3 to assess the suggested model's accuracy. Also, the next part will display the accuracy results associated to them.

5.3 Metrics and Results

The following figures and tables are report of the experimental findings used to assess the efficiency of the suggested model for business process outcome prediction. Here, four different experiment parameters setting and their related accuracy results are shown.

the hyperparameters: the quantity of neurons, the activation function, the optimizer, the learning rate, the batch size, and the epochs.
layers: the quantity of hidden layers.

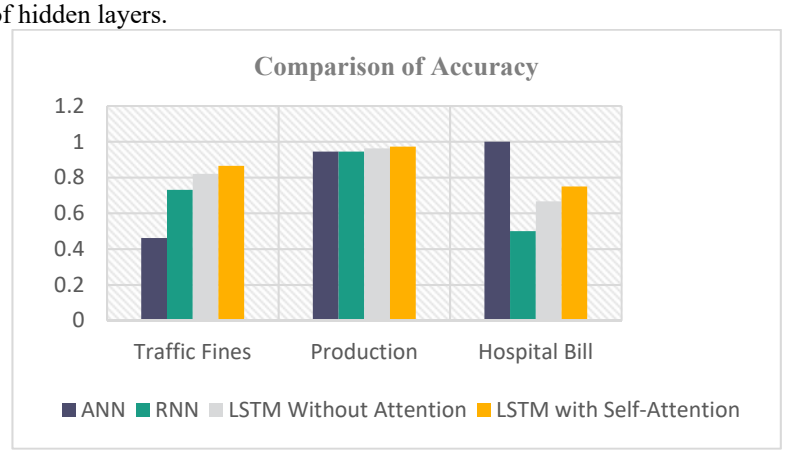

Figure 4 Accuracy Results for Experiment Parameter Setting (1)

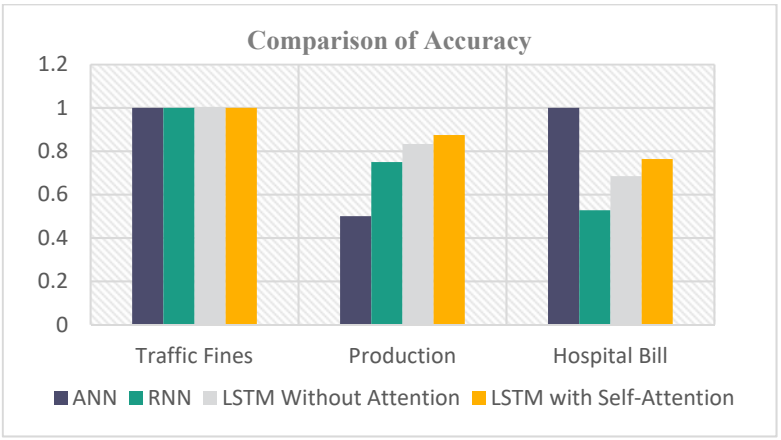

Figure 5 Accuracy Results for Experiment Parameter Setting (2)

As the above- mentioned the experiment parameter setting is set as an initial setting. Then, tuning the hyper parameters by lowering the quantity of neurons and rising learning rate. The difference results can be seen in the figures 4, 5.

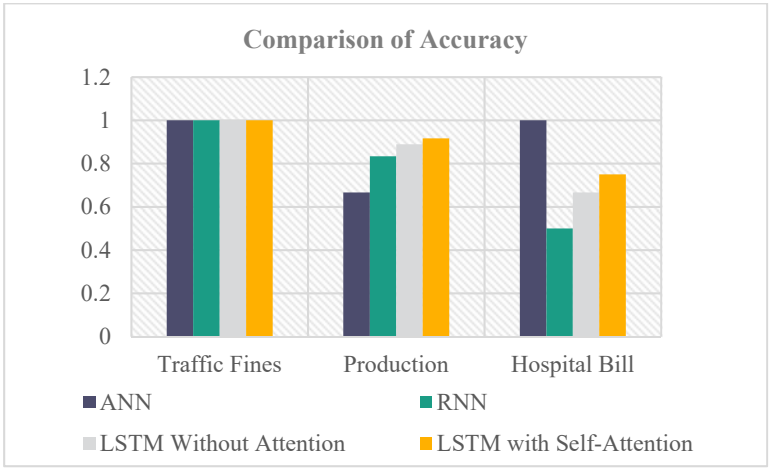

Figure 6 Accuracy Results for Experiment Parameter Setting (3)

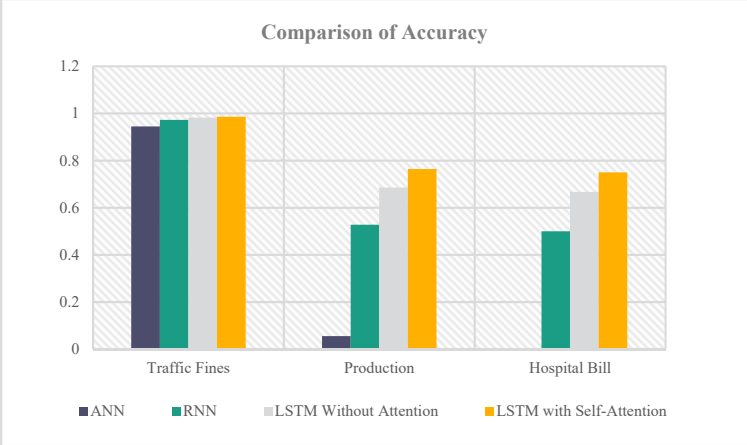

Figure 7 Accuracy Result of Proposed Models with Experiment Parameters Setting (4)

The Figure 6 is related to the changing of optimizer or updater function with Nesterovs and learning rate value. The suggested LSTM with self-attention model and the original LSTM model's batch size and number of hidden layers' effects are discussed in Figure 7.

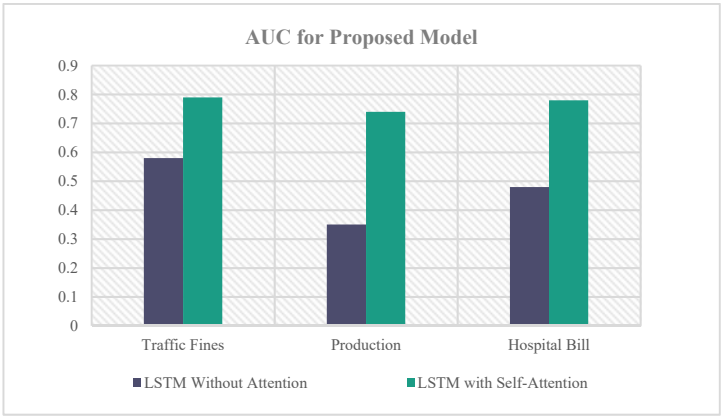

Figure 8 AUC of Proposed Model by Comparing with Attention and Without Attention

The usage metric to reflect the performance of prediction is applied to assess the precision of the suggested model. The selection of the AUC, or area under the curve ROC metric, which is employed in classification analysis to ascertain which of the used predicts classes best, is done to evaluate the proposed models for the chosen datasets and the binary character of the targets. The accuracy value and the AUC are computed on the validation set at the end of each epoch. For AUC, it only compares the proposed self-attention-based LSTM model with the original LSTM model for the selected datasets, which is shown in Figure 8.

6. Conclusion and future work

For a prediction to be correct, a given dataset needs a specific set of hyperparameters. Users, however, find it difficult to pick which hyperparameter to utilize due to the abundance of them. There is no conclusive response as to the optimal number of layers, neurons, or optimizers for each dataset. To develop a model from a specific dataset, hyperparameter tweaking is crucial to identifying the best potential several hyperparameter settings. To sum up, these have presented self-attention-based LSTM model for predicting outcomes from business process execution. It is implemented with java based DL4J framework and experimented on 3 datasets. From these results, it is at first, observed that the proposed attention-based LSTM model achieves higher accuracy than LSTM model without attention. Furthermore, the AUC value of the proposed model is range 0.7 to 0.8, so the proposed model is considered acceptable. In addition, with various deep learning architectures and training settings, such as learning rate and various dropout rates will be examining with other attention mechanisms such as global attention, hierarchical attention with different datasets, which can be implemented as the future work for this research.

Conflict of interest

All authors declare no conflicts of interest in this paper.

REFERENCES

[1] Rama-Maneiro, E., Vidal, J., & Lama, M. (2021). Deep learning for predictive business process monitoring: Review and benchmark. IEEE Transactions on Services Computing.

[2] Bukhsh, Z. A., Saeed, A., & Dijkman, R. M. (2021). Process transformer: Predictive business process monitoring with transformer network. arXiv preprint arXiv:2104.00721.

[3] De Smedt, J., & De Weerdt, J. (2020). Predictive Process Model Monitoring using Recurrent Neural Networks. arXiv preprint arXiv:2011.02819.

[4] Metzger, A., Leitner, P., Ivanovic, D., Schmieders, E., Franklin, R., Carro, M., ... & Pohl, K. (2014). Comparing and combining predictive business process monitoring techniques. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45(2), 276-290.

[5] Schwegmann, B., Matzner, M., & Janiesch, C. (2013). A method and tool for predictive event-driven process analytics.

[6] Milani, F., & Di Francescomarino, C. A Literature Review on Predictive Monitoring of Business Processes.

[7] Di Francescomarino, C., Ghidini, C., Maggi, F. M., & Milani, F. (2018, August). Predictive process monitoring methods: Which one suits me best? In International conference on business process management (pp. 462-479). Cham: Springer International Publishing.

[8] I Verenich, M Dumas, M La Rosa, FM Maggi, C Di Francescomarino," A general framework for predictive business process monitoring", International Conference on Advanced Information Systems Engineering, 186-202, 2016. 5, 2016.

[9] Teinemaa, I., Dumas, M., Leontjeva, A., & Maggi, F. M. (2018). Temporal stability in predictive process monitoring. Data Mining and Knowledge Discovery, 32, 1306-1338.

[10] Teinemaa, I., Dumas, M., Rosa, M. L., & Maggi, F. M. (2019). Outcome-oriented predictive process monitoring: Review and benchmark. ACM Transactions on Knowledge Discovery from Data (TKDD), 13(2), 1-57.

[11] J. Evermann, J.-R. Rehse, and P. Fettke. "A Deep Learning Approach for Predicting Process Behaviour at Runtime", pages 327–338. Springer International Publishing, Cham, 2017.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modelling," arXiv preprint arXiv:1412.3555, 2014.

[13] K. Cho, B. Van Merri¨enboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv preprint arXiv: 1409.1259, 2014.

[14] M.-Chamorro AE, Resinas M, Ruiz-Cortes A (2017) Predictive monitoring of business processes: a survey. IEEE Transactions on Services Computing.

[15] M. Dumas, Marcello LaRosa, JanMendling, and HajoA. Reijers. 2018. Fundamentals of Business Process Management (2nd ed.). Springer.

[16] M. Polato, A. Sperduti, A. Burattin, and M. de Leoni. "Data-aware remaining time prediction of business process instances." In 2014 International Joint Conference on Neural Networks (IJCNN), pages 816–823. IEEE, July 2014.

[17] M. Hinkka1, Teemu Lehto1;2, Keijo Heljanko1;3, and Alexander Jung, "Classifying Process Instances Using Recurrent Neural Networks", 2018.

[18] Navarin, N., Vincenzi, B., Polato, M., & Sperduti, A. (2017, November). LSTM networks for data-aware remaining time prediction of business process instances. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-7). IEEE.

[19] N. Tax, I. Verenich, M. L. Rosa, and M. Dumas, "Predictive business process monitoring with LSTM neural networks," in International Conference on Advanced Information Systems Engineering (CAiSE). Springer, 2017, pp. 477–492.

[20] T. Hnin, K. K. Oo, "Deep Learning for Predictive Process Behavior ", 16th International Conference on Computer Application (ICCA), Yangon, Myanmar, 2018, pp.167-170.

[21] T. Hnin, K. K. Oo, "Attention Based LSTM with Multi Tasks Learning for Predictive Process Monitoring", Proceedings of 2019 the 9th International workshop on computer science and Engineering WCSE - 2019 SPRING Yangon, Myanmar, February 27- March 1, 2019, pp.165-170.

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J. Jones, L. Gomez, A. N. Kaiser, L. Polosukhin, Attention is all you need. In: Proc. of NeuroIPS (2017).

[23] Wolf, T. Chaumond, J. Debut, L. Sanh, V. Delangue, C. Moi, A. Cistac, P. Funtowicz, M. Davison, J. Shleifer, Transformers: State-of-the-art natural language processing. In: Proc. of EMNLP (2020).

[24] J. Wang, D. Yu, C. Liu, X. Sun," Predicting Outcomes of Business Process Executions Based on LSTM Neural Networks and Attention Mechanism", Research Square, April 26th, 2021.

[25] Brauwers, G., & Frasincar, F. (2021). A general survey on attention mechanisms in deep learning. IEEE Transactions on Knowledge and Data Engineering.

[26] Tello-Leal, E., Roa, J., Rubiolo, M., & Ramirez-Alcocer, U. M. (2018, November). Predicting activities in business processes with LSTM recurrent neural networks. In 2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K) (pp. 1-7). IEEE.

Authors Profile

Thuzar Hnin received her M.I.Sc. degree in computer science from University of Computer Studies, Yangon, Myanmar in 2002. She is an associate professor of Faculty of Information Science at University of Computer Studies, Yangon, Myanmar. Her research interests mainly include data mining, artificial intelligence, machine learning and intelligent planning.

Tin Thein Thwel, Ph.D. (IT) is a Professor at University of Computer Studies, Yangon (UCSY), Yangon, Myanmar. She received her PhD in Information Technology from UCSY, Myanmar in 2010. Her research interest includes data deduplication, cyber security, data mining, data science, information retrieval, artificial intelligent, machine learning.