



Business Process Remaining Time Prediction Based on Bidirectional QRNN with Attention Mechanism

Na Guo ¹, Ting Lu ¹, Cong Liu ^{2*}, Xingrong Xu ¹, Qingtian Zeng ³

¹ School of Computer Science and Technology, Shandong University of Technology, Zibo 255000, China.

² NOVA Information Management School, Nova University of Lisbon, Lisbon 1070-312, Portugal.

³ College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China.

Abstract

Business process prediction is essential for monitoring workflows and ensuring service quality. A key task in this area, remaining time prediction, focuses on estimating process duration and has been extensively studied. While Long Short-Term Memory (LSTM) networks are widely adopted, their limited parallelization and sequential modeling capabilities constrain performance. To address these limitations, we propose a remaining time prediction approach based on a bidirectional Quasi-Recurrent Neural Network (QRNN) with an attention mechanism. Specifically, the bidirectional QRNN is employed to construct the prediction model, while the attention mechanism enhances its ability to extract feature information. Next, a transfer training iteration strategy based on different trace prefix lengths is designed to address the imbalance in trace lengths. Then, a Word2Vec-based event representation learning approach is introduced to generate similarity vector of adjacent events, further improving prediction accuracy. Finally, using five publicly real-life event logs, the proposed approach is evaluated against state-of-the-art approaches. Experimental results demonstrate that it improves average prediction accuracy by nearly 15% while reducing average model training time by approximately 26%.

Keywords:

Process Mining;
Remaining Time Prediction;
Quasi-Recurrent Neural Network;
Representation Learning.

Article History:

Received:	22	September	2025
Revised:	19	February	2026
Accepted:	12	March	2026
Published:	01	April	2026

1- Introduction

Predictive process monitoring [1], a key research direction within process mining [2, 3], aims to accurately predict the future execution states, such as the next activity, and remaining time [4], by analyzing the ongoing process instances. Remaining time prediction [5] is a fundamental task in predictive process monitoring, aiming to estimate the time required to complete an ongoing process instance.

Traditional approaches to remaining time prediction are model-based [6], relying on explicit process models such as transition systems [7] or stochastic Petri nets [8] discovered from historical event logs. Although interpretable, these models often failed to handle the noise and variability inherent in real-life process data. With the advent of deep learning, researchers have proposed data-driven approaches that learn temporal and contextual dependencies directly from event logs. In particular, Long Short-Term Memory (LSTM) neural networks [9] have been widely applied, demonstrating superior predictive performance compared to traditional models. More recently, attention-based and Transformer deep learning architectures [10, 11] have been introduced to capture long-range dependencies and contextual information more effectively.

* **CONTACT:** cliu@novaims.unl.pt

DOI: <https://doi.org/10.28991/ESJ-2026-010-02-01>

© 2026 by the authors. Licensee ESJ, Italy. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<https://creativecommons.org/licenses/by/4.0/>).

However, the process of remaining time prediction based on deep learning techniques is still in its infancy, and there are still the following limitations. (1) The applicability of deep learning techniques often depends on high-quality training data. In business processes, significant differences in trace lengths across different process variants lead to an imbalanced distribution of trace prefix lengths. Specifically, short trace prefixes are far more prevalent than long ones, which can reduce the accuracy of remaining time prediction. (2) Most existing approaches employ LSTM to construct remaining time prediction models. However, LSTM exhibits limited modeling capability for sequential data and suffers from slow training speed when processing large-scale datasets. Therefore, further research is needed to enhance prediction accuracy while improving the model training efficiency.

To handle the above limitations, we propose a business process remaining time prediction approach. The main innovations include the following.

- (1) A remaining time prediction model based on bidirectional QRNN with an attention mechanism is constructed, which uses a matrix convolution operation to improve the parallel computing ability and training efficiency and introduces an attention mechanism to highlight the influence of important features on the prediction results.
- (2) A transfer training iteration strategy based on different trace prefix lengths is designed to address and enhance the quantitative variability and correlation among different-length trace prefixes in process instances.
- (3) A Word2Vec-based event representation learning approach is proposed to enhance the correlation between events and improve the prediction accuracy through the high-quality vectorized representation of events.

The rest of this paper is organized as follows. Section 2 presents a review of related work. Section 3 introduces the background knowledge. Section 4 presents the proposed approach based on a bidirectional quasi-recurrent neural network with attention. Section 5 evaluates the effectiveness and applicability of the proposed approach using five real-life event logs. Section 6 concludes this paper and outlines future research directions.

2- Related Works

The business process remaining time prediction is a valuable research area in process mining, enabling the optimization of business system performance and providing a reliable decision-making basis for managers [12]. van der Aalst et al. [7] employed a multi-level abstract transition system to record all possible states of process instances in event logs, annotating each state with statistical time information to predict remaining time. Rogge-Solti et al. [8] simulated ongoing process instances using a stochastic Petri net discovered from event logs to predict remaining time. However, traditional process mining techniques face limitations in remaining time prediction, especially when dealing with large-scale event logs, where achieving high prediction accuracy remains a significant challenge.

With the widespread adoption of machine learning techniques, researchers have integrated them into business process remaining time prediction [13]. Folino et al. [14] proposed a logic-based predictive clustering framework that first clusters traces and then builds a local predictor for each cluster to better support post-hoc analysis [15]. Polato et al. [16] introduced a data-aware transition system based on [7], leveraging Bayesian classification to establish a support vector machine for existing state nodes to predict remaining time. Verenich et al. [17] abstracted business processes into process trees and trained regression models on each process tree node to estimate remaining time. However, existing machine learning techniques often struggle to capture rich and discriminative information from process data [18].

In recent years, deep learning techniques have been increasingly applied to process remaining time prediction [19, 20]. Evermann et al. [21] introduced an initial application of deep learning with LSTM neural networks for process prediction. Nguyen et al. [22] proposed a novel feature representation approach to enhance the prediction capability of deep learning models. Ni et al. [23] further improved prediction accuracy by incorporating an attention mechanism based on [4]. Khan et al. [24] employed memory-enhanced neural networks to address complex challenges in business process remaining time prediction. Pasquadibisceglie et al. [25] proposed a data engineering approach that transformed temporal information into an abstract spatial image structure, subsequently using a convolutional neural network to predict remaining time. Taymouri et al. [26] leveraged generative adversarial networks (GANs) to predict remaining time while generating event suffixes. Bukhsh et al. [10] introduced an advanced representation approach that captures long-term dependencies in input sequences using the transformer model, enhancing the remaining time prediction accuracy. Amiri Elyasi et al. [27] transformed event logs into graph datasets and leveraged graph-oriented data to train Process Graph Transformer Networks for predicting the remaining time of business process instances. Liu et al. [28] evaluated the impact of applying different activity encoding techniques as input on the accuracy of remaining time prediction in deep learning models. Guo et al. [29] proposed a feature-informed cascade prediction model based on LSTM, designed to improve prediction accuracy through the separate processing of features. These advances demonstrate the effectiveness of deep learning techniques for remaining time prediction. Nevertheless, opportunities remain to further enhance prediction accuracy, indicating the need for more robust and tailored modeling approaches.

3- Background Knowledge

In this section, we briefly revisit the basic concepts of event logs and the task of process remaining time prediction.

3-1-Event Logs

Definition 1 (Event): An event represents an activity execution instance within a business system and can be defined as a tuple $e = (a, id, T_{start}, T_{end}, P_1, \dots, P_m)$. Where a represents the executed activity, id is the unique identifier of the event, T_{start} and T_{end} represent the start time and end time, and P_1, \dots, P_m are additional event attributes, such as resource and cost.

Definition 2 (Trace): A trace is a non-empty, finite sequence of events, represented as $\sigma = (e_1, \dots, e_{|\sigma|})$, where $|\sigma|$ represents the number of events in the trace.

Definition 3 (Trace Prefix): Trace prefix refers to the first k events in a trace σ , expressed as $\sigma^{(k)} = (e_1, \dots, e_k)$, where $1 \leq k \leq |\sigma|$. Trace prefix remaining time $RT(\sigma^{(k)}) = e_{|\sigma|}.T_{end} - e_k.T_{end}$.

Definition 4 (Process Instance): Process Instance is a business complete execution from start to end, which can be expressed as a triplet $z = \{Cid, \sigma, P_{\{1,2,\dots,m\}}\}$. Among them, Cid represents the unique identifier of the process instance, σ represents the activity trace, $P_{\{1,2,\dots,m\}}$ represents other attributes.

Definition 5 (Event Log): Event Log is a set of process instances, which records the business system execution, expressed as $L = \{\sigma_1, \sigma_2, \dots, \sigma_{|l|}\}$, where $|l|$ represents the number of process instances.

Definition 6 (Remaining Time Prediction): Given the event space X , the traces and trace prefixes space in the process instance are expressed as X^* . Remaining time prediction aims at predicting the remaining time of the process instance trace prefix through the established prediction model $f = Y^* \rightarrow R^+$.

3-2-Business Process Remaining Time Prediction

Business process remaining time prediction using deep learning techniques consists of two phases: training and application. In the training phase, the prediction model f is trained by constructing the training dataset from the event log. The training dataset is generated by intercepting different trace prefixes lengths in process instances, and the specific process is shown in algorithm 1. Specifically, each trace σ in the event log L is traversed, and the trace prefix $\sigma^{(k)}$ and the corresponding remaining time $RT(\sigma, k)$ are intercepted according to the specified trace prefix length $k \in [k_{min}, k_{max}]$ as the training subset D_k . Finally, all training subsets are combined to generate the complete training dataset D . In the application phase, the trained model f is used to predict the remaining time of ongoing process instances.

Algorithm 1. Constructing labeled data

Input: event log L , minimum trace prefix length k_{min} , maximum trace prefix length k_{max}
Output: training dataset $D = \{D_{k_{min}} \cup \dots \cup D_{k_{max}}\}$

```

1:  $D_k \leftarrow \emptyset$ ;
2: FOR  $\sigma \in L$  DO
3:   FOR  $k \in [k_{min}, k_{max}]$  DO
4:     IF  $k < \text{len}(\sigma)$  THEN
5:        $D_k \leftarrow \{(\sigma^{(k)}, RT(\sigma, k))\}$ 
6:     END IF
7:   RETURN  $D = \{D_{k_{min}} \cup \dots \cup D_{k_{max}}\}$ 
8:   END DO
9: END DO
```

4- Remaining Time Prediction Approach-based on Bidirectional QRNN with Attention Mechanism

This section first gives an overview of the approach and then explains in detail the remaining time prediction approach.

4-1-An Overview of the Approach

Figure 1 shows an overview of the approach that consists of the following three steps:

Step 1: Training datasets generation. Taking an event log as input, the process remaining time training datasets with different trace prefix lengths are generated according to Algorithm 1.

Step 2: Event representation learning. All events in the training datasets are extracted, and their representation vectors are learned using the Word2Vec encoding technique.

Step 3: Construction and training of the remaining time prediction model. Bidirectional QRNN and attention mechanism are used to construct the process remaining time prediction model, and the event representation is used as model input. The prediction model is trained by using the trace prefix training iteration strategy with different lengths.

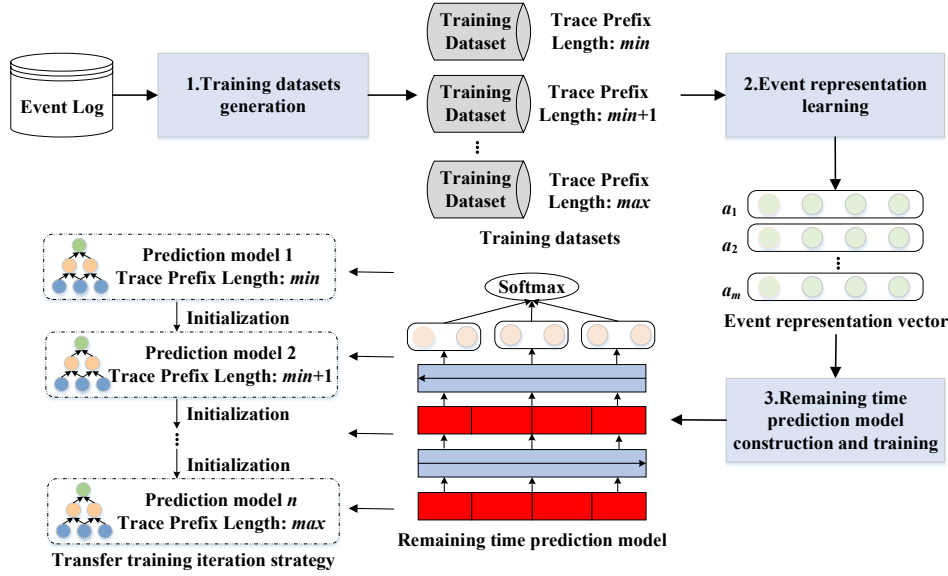


Figure 1. An approach overview

4-2- Word2Vec-based Event Representation Learning

Deep learning techniques for business process remaining time prediction commonly rely on one-hot encoding to represent events. However, one-hot encoding suffers from a semantic gap, which can negatively affect the prediction accuracy. To address this limitation, a Word2Vec-based event representation learning approach is proposed to generate high-quality, context-aware event representation vector. These vectors serve as inputs to the prediction model, thereby enhancing the remaining time prediction accuracy. Activity and duration are key components of event and are used as the primary features for vectorization. Since duration is a continuous variable, it must first be discretized to enable more refined and meaningful representation of each activity prior to vectorization. The discretization process is defined as follows.

$$\tilde{t}(e) = \left\lfloor \frac{t(e) - t_{\text{MIN}}(\text{EndTime-StartTime})(e.a)}{t_{\text{MAX}}(\text{EndTime-StartTime})(e.a) - t_{\text{MIN}}(\text{EndTime-StartTime})(e.a)} \cdot N \right\rfloor \quad (1)$$

As shown in Equation 1, $t(e)$ represents the duration of event e , $t_{\text{MIN}}(\text{EndTime-StartTime})(e.a)$ and $t_{\text{MAX}}(\text{EndTime-StartTime})(e.a)$ represent the shortest and longest duration of all activities, N represents the number of discretization intervals for the duration of each activity. Then the activity is combined with the discrete duration to represent the event.

We use continuous bag-of-words model (CBOW) in Word2Vec [30] to learn event representation vector. Specifically, the binary encodings of n context events $\{e_{c-n}, \dots, e_{c-1}, e_{c+1}, \dots, e_{c+n}\}$ of the current event e_c are used as input to the training model, while the one-hot encoding of the event e_c serves as the output. After training, the weight matrix between the input layer and the hidden layer is the event representation vector matrix.

4-3- Bidirectional QRNN with Attention Mechanism

In LSTM networks, each time step depends on the output of the previous time step, limiting effective parallel computation when dealing with large-scale sequence data and reducing the modeling ability of sequence data. QRNN [31] addresses this issue by using alternating convolutional layers, which fully utilize the sequence information and enable parallel processing across time steps [32]. In addition, QRNN simplifies the LSTM structure by retaining only the forgetting and output gates, thereby reducing computational complexity [33]. The main components of QRNN are the convolutional layer and f_0 -pooling layer. The convolutional layer extracts feature information from the input sequence and performs convolution on the gate functions, while the f_0 -pooling layer processes the output of the convolutional layer to reduce the number of features (see Figure 2).

The sequence $X = \{x_1, x_2, \dots, x_T\}$ with the length T input to the convolutional layer is convolved by filters with the number w and the width r in the time dimension. The f_0 -pooling layer receives the convolution results as input, and its cell structure incorporates both the forgetting and output gates. At time t , the input sequence is $\{x_{(t-r+1)}, \dots, x_t\}$, and the output z_t , forgetting gate f_t and output gate o_t are calculated as follows.

$$z_t = \tanh(W_z^1 x_{t-r+1} + W_z^2 x_{t-r+2} + \dots + W_z^r x_t) \quad (3)$$

$$f_t = \text{Sigmoid}(W_f^1 x_{t-r+1} + W_f^2 x_{t-r+2} + \dots + W_f^r x_t) \quad (4)$$

$$o_t = \text{Sigmoid}(W_o^1 x_{t-r+1} + W_o^2 x_{t-r+2} + \dots + W_o^r x_t) \quad (5)$$

As shown in Equations 3 to 5, where W_z , W_f , and W_o represent weight matrices, while Sigmoid and tanh denote the activation functions. As the filter width r increases, the model is capable of capturing more N-gram features, thereby enhancing its ability to extract local patterns from the input sequence.

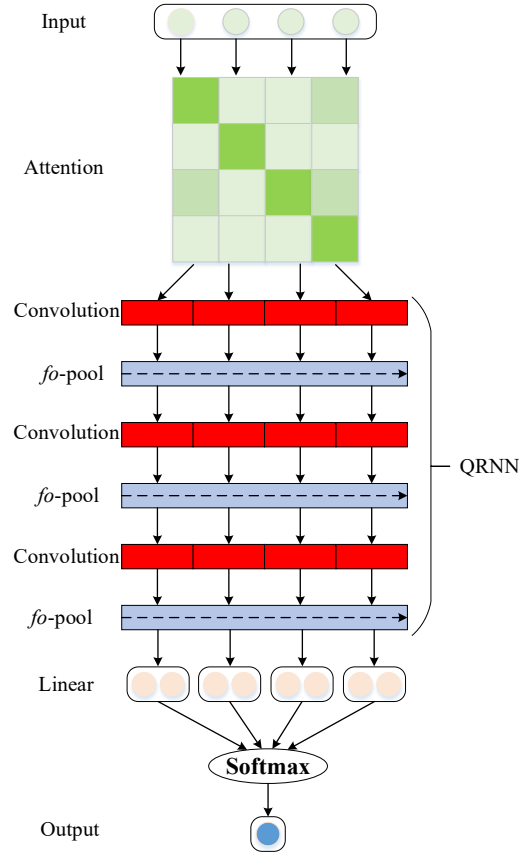


Figure 2. Architecture of bidirectional QRNN with attention mechanism

Finally, the memory cell state c_t at moment t is updated, and the QRNN output h_t is calculated from the output gate o_t and the memory cell state c_t . The calculation formula is as follows.

$$c_t = f_t \cdot c_{t-1} + (1 - f_t) \cdot z_t \quad (6)$$

$$h_t = o_t \cdot c_t \quad (7)$$

As shown in Equation 6, c_{t-1} represents the memory cell state at the previous moment.

To fully capture the context and effectively distinguish the importance of different event locations, we use a bidirectional quasi-recurrent neural network with an attention mechanism (Att-Bi-QRNN) to construct the remaining time prediction model. The model consists of the following key modules:

(1) Bidirectional QRNN-based context encoding

Compared to a standard QRNN, the bidirectional QRNN can fully account for the correlation between different events in the trace prefixes. At the moment t , the output hidden vectors from the forward and backward QRNNs are denoted as \vec{h}_t and \overleftarrow{h}_t . These hidden vectors are concatenated to form the final context encoding $H_t = [\vec{h}_t, \overleftarrow{h}_t]$, capturing the bidirectional dependencies of events.

(2) Attention-based trace prefixes encoding

Incorporating the attention mechanism, the weight a_t for the context encoding H_t is computed to reflect the relative importance of each event in the trace prefix for the remaining time prediction. The trace prefix encoding is then calculated by applying these attention weights to the context encoding as follows.

$$v = \sum_t a_t \cdot H_t \quad (8)$$

4-4- Transfer training Iteration Strategy based on Different Trace Prefix Lengths

Compared to training a single remaining time prediction model using all data, training multiple models based on different trace prefix lengths can enhance the pertinence of prediction model. However, this approach presents two key challenges: (1) when the event log is divided into multiple training datasets based on varying trace prefix lengths, longer prefixes result in smaller training datasets. Since deep learning models typically require large-scale training data, this scarcity can negatively impact model performance; and (2) longer trace prefixes may exclude certain process variants present in shorter trace prefixes, potentially overlooking correlations between some events when training on them independently.

To address the above issues, we design a transfer training iteration strategy based on different trace prefix lengths, with the detailed steps outlined in Algorithm 2. Specifically, the range of trace prefix lengths $[min, max]$ is defined, where min and max represent the shortest and longest trace prefix length in the event log after filtering out infrequent long traces. The infrequent long traces may lead to overfitting, which can negatively impact the remaining time prediction accuracy. Using Algorithm 1, multiple training datasets $D_{k.min}, \dots, D_{k.max}$ are generated. The remaining time prediction model $F_{k.min}$ is trained on $D_{k.min}$. The training parameters are shared with $F_{k.min+1}$, and the model is fine-tuned by $D_{k.min+1}$. This process is iterated until $F_{k.max}$ is obtained. This strategy allows the model to fully capture the internal correlations within different trace prefix lengths, addressing the issue of small-scale for longer prefixes and improving the remaining time prediction accuracy.

Algorithm 2. Transfer training iteration strategy based on different trace prefix lengths

Input: Training datasets $D_{k.min} \cup \dots \cup D_{k.max}$, Iterations n

Output: Remaining time prediction model for different trace prefix lengths $\{F_{k.min}, F_{k.min+1}, \dots, F_{k.max}\}$

```

1: FOR  $i \in [1, n]$  DO
2:    $F_{k.min} \leftarrow \text{Train}(D_{k.min})$ 
3:   FOR  $k \in [k_{min+1}, k_{max}]$  DO
4:      $F_{(k.min+1, \dots, k, \dots, max)} \leftarrow \text{Train}(D_{(k.min+1, \dots, k, \dots, max)}, F_{(k.min, \dots, k-1, \dots, max-1)})$ 
5:   RETURN  $\{F_{k.min}, F_{k.min+1}, \dots, F_{k.max}\}$ 
6: END DO
7: END DO

```

5- Experimental Evaluation

This section shows the experimental setup and results analysis of the remaining time prediction approach based on bidirectional quasi-recurrent neural network with attention mechanism. Our approach is implemented in Pytorch. The computer configuration used is: PC Intel Core i5-10400F 2.90 GHz, NVIDIA GeForce RTX 2070 SUPER.

5-1- Data Set

Our experiments use five real-life event logs obtained from the 4TU Research Centre. Among them, BPIC_2012_A, BPIC_2012_O, and BPIC_2012_W logs, sourced from a Dutch financial institution, record the loan application approval process, the change process of loan approval status, and the change process of resource allocation in loan approval, respectively. The Helpdesk log captures back-office ticket management processes from an Italian software company, while the Hospital_Billing log documents the discharge billing procedures in the EPR system of a regional hospital. Table 1 shows a statistical overview of five event logs.

Table 1. Statistics of event log

Datasets	Traces	Events	Activities	Max length of trace	Min length of trace
BPIC_2012_A [34]	13087	73022	10	10	3
BPIC_2012_O [34]	5015	41728	7	39	4
BPIC_2012_W [34]	9658	147450	6	153	1
Helpdesk [35]	3804	13710	9	14	1
Hospital_Billing [36]	100000	451359	18	217	1

5-2-Experimental Settings

To evaluate the remaining time prediction accuracy, the mean absolute error (MAE) is applied and computed as follows.

$$\text{MAE}(f) = \sum_{(\sigma^{(k)}, \text{RemainTime}(\sigma, k)) \in D} |f(\sigma^{(k)}) - \text{RemainTime}(\sigma, k)| \quad (9)$$

As shown in Equation 9, $\sigma^{(k)}$ represents the trace prefixes, $f(\sigma^{(k)})$ represents the prediction value, and $\text{RemainTime}(\sigma, k)$ represents the true value. A smaller MAE value indicates a higher prediction accuracy.

In the event representation learning phase, the number of discretization intervals N is set to 10 [23]. In the prediction model construction phase, the filter width r of the convolutional layer is set to 2. In the prediction model training phase, the hyperparameters are configured as follows: batch size is set to 512, epoch is set to 150, learning rate is 0.01, and the Adam optimization algorithm is used. The experiment evaluates the remaining time prediction model using 5-fold cross-validation. Each dataset is randomly divided into five equal parts, with four parts used for training and the remaining part used for testing. This process is repeated five times, ensuring that each subset serves as the test set once. The average MAE across the five runs is then calculated and reported as the final evaluation result.

5-3-Experimental Results of Remaining Time Prediction

To demonstrate the effectiveness of the QRNN-based prediction model in the remaining time prediction, we compare it with the attention-based bidirectional recurrent neural network approach proposed in Ni et al. [23], using the same settings for the number of hidden layers and nodes. The experimental results are shown in Table 2.

Table 2. Results of process remaining time prediction models

Models	Helpdesk	Hospital_Billing	BPIC_2012_A	BPIC_2012_O	BPIC_2012_W
LSTM [23]	3.542	42.050	3.588	7.993	8.021
Bi-LSTM [23]	2.952	41.515	4.600	8.407	7.221
Att-Bi-LSTM [23]	2.677	36.691	3.895	7.324	6.153
Trans-Att-Bi-LSTM [23]	3.357	33.201	3.489	5.858	5.826
GRU [23]	3.425	47.400	4.888	7.394	7.753
Bi-GRU [23]	3.222	37.157	4.461	8.144	7.334
Att-Bi-GRU [23]	3.303	33.201	3.512	7.306	6.338
Trans-Att-Bi-GRU [3]	3.299	32.187	3.438	5.863	5.821
QRNN	3.199	36.611	3.543	6.779	7.516
Bi-QRNN	2.884	35.248	3.992	6.757	7.128
Att-Bi-QRNN	2.626	32.101	3.200	6.276	6.007
Trans-Att-Bi-QRNN	2.423	31.436	2.373	5.158	5.275

Comparing the three types of prediction models based on QRNN, LSTM, and GRU, it is evident that QRNN achieves overall higher prediction accuracy across the five event logs, making it more suitable for the remaining time prediction task. With the introduction of the bidirectional mechanism, Bi-QRNN outperforms QRNN in terms of lower MAE values, demonstrating that bidirectional modeling effectively captures the correlations between events and enhances prediction accuracy. Building on Bi-QRNN, Att-Bi-QRNN incorporates an attention mechanism to differentiate the influence of individual events, further reducing the MAE. This indicates that the attention mechanism helps the model focus on key events, leading to additional improvements in prediction accuracy. Specifically, across the five event logs, the MAE of Bi-QRNN is approximately 10% lower on average than that of Bi-LSTM and Bi-GRU, while the MAE of Att-Bi-QRNN is nearly 4% lower than that of Att-Bi-LSTM and Att-Bi-GRU. These results confirm the effectiveness and practicality of using QRNN for remaining time prediction. Trans-Att-Bi-QRNN is a prediction model obtained by adopting the transfer training strategy. Compared to the best-performing model in Ni et al. [23], the proposed Trans-Att-Bi-QRNN, which adopts the QRNN architecture, achieves an average 15% reduction in MAE across the five event logs. This underscores the superiority of the proposed approach in remaining time prediction.

To demonstrate the effectiveness of the training iteration strategy based on different trace prefix lengths, we compare Trans-Att-Bi-QRNN with the prediction model Diff-Att-Bi-QRNN trained separately on different trace prefix datasets. Figure 3 shows the MAE values for both models across different trace prefix lengths. It can be seen that the MAE values of Trans-Att-Bi-QRNN is lower than that of Diff-Att-Bi-QRNN in all logs, indicating that the training iteration strategy improves remaining time prediction accuracy. Furthermore, for Helpdesk and Hospital_Billing logs, Trans-Att-Bi-QRNN demonstrates significant advantages, particularly for longer trace prefixes. This highlights that the training iteration strategy can overcome the challenges by small-scale and effectively capture correlations between different trace

prefix lengths. In addition, the short-term prediction error is usually large at the beginning of the process. As the trace prefix length increases, the prediction error gradually decreases and stabilizes. It does not increase sharply with the reduction in data size, which can be attributed to the availability of more prior knowledge.

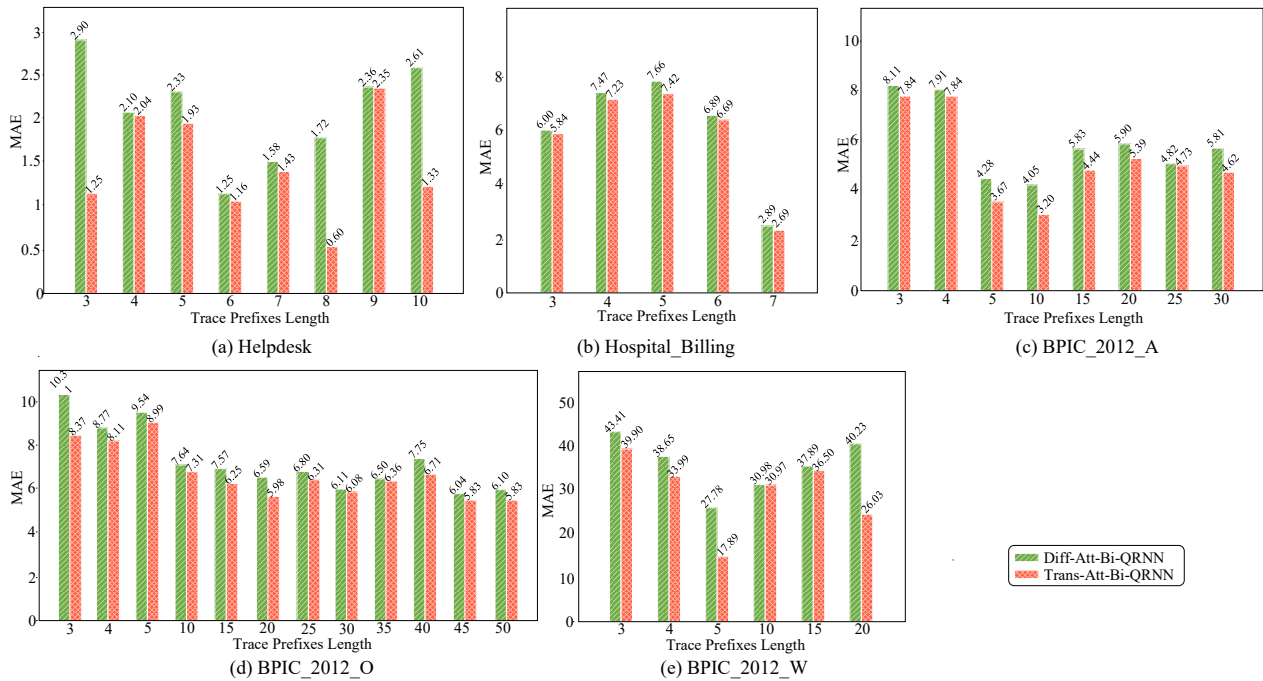


Figure 3. Results of training iteration strategy based on different trace prefix length.

To assess the impact of the Word2Vec-based event representation learning approach on remaining time prediction accuracy, we use Trans-Att-Bi-QRNN-OH, which employs one-hot event encoding, as a baseline for comparison with Trans-Att-Bi-QRNN. The results are shown in Figure 4. Across all logs, the MAE value of Trans-Att-Bi-QRNN is 3% lower than that of Trans-Att-Bi-QRNN-OH, demonstrating the positive effect of the event representation approach on remaining time prediction.

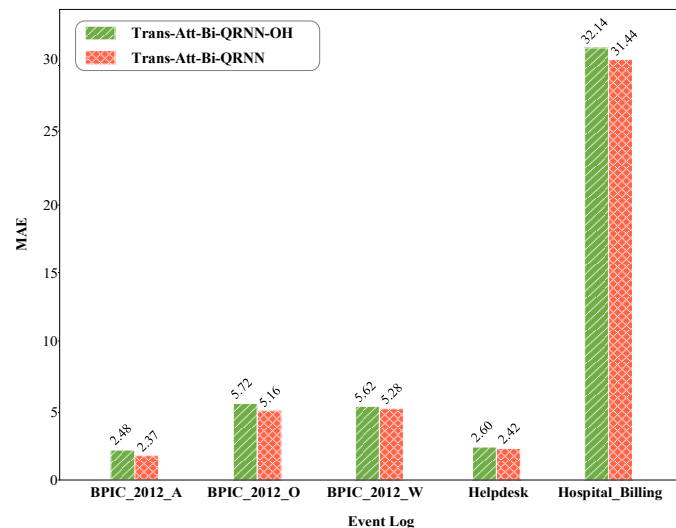


Figure 4. Results of Word2Vec-based event representation learning.

5-4-Comparative Analysis of Prediction Model Performance

We conducted a comparative experiment to evaluate the training time (in seconds) of all prediction models, and the results are shown in Figure 5. It can be seen that whether the Word2Vec-based event representation learning approach is applied or not and the training iteration strategy is trained based on different trace prefix lengths, the average training time of QRNN-based prediction models is about 26% shorter than that of LSTM-based or GRU-based prediction models in five logs. These results demonstrate that the QRNN-based prediction model achieves significantly higher training efficiency, thereby enhancing the overall performance.

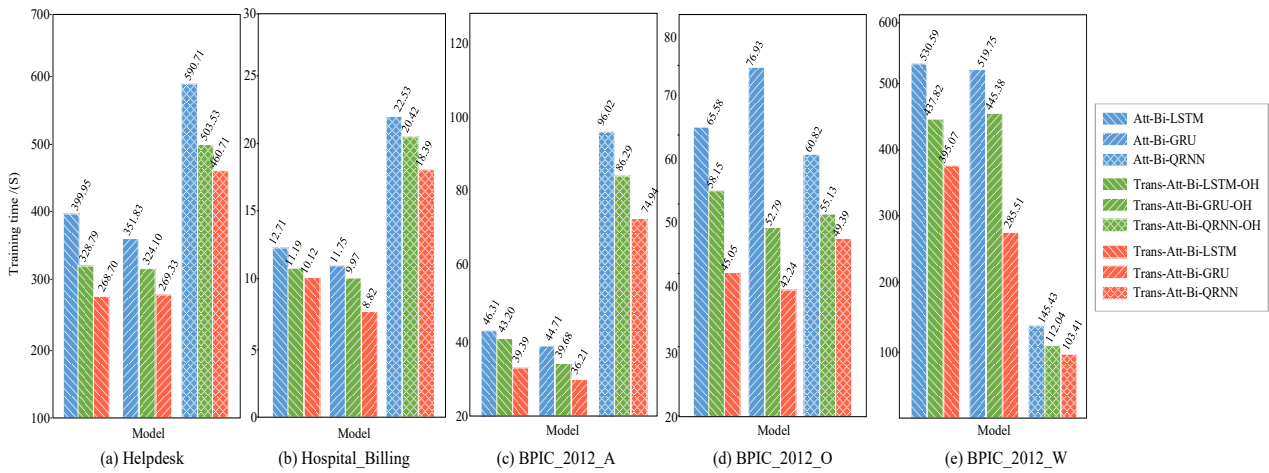


Figure 5. Comparison of prediction model training time

6- Conclusion

Process remaining time prediction based on deep learning techniques is still in its infancy. Current approaches depend on balanced, high-quality data and often suffer from imbalanced trace lengths. Moreover, LSTM-based models face limitations in capturing long dependencies and training efficiency, highlighting the need for more accurate and efficient approaches. In this paper, we proposed a process remaining time prediction approach based on bidirectional quasi-recurrent neural network (QRNN) with attention mechanism. The use of QRNN improves the training efficiency of the prediction model, while the integration of bidirectional and attention mechanism enables the model to capture contextual dependencies and emphasize important features influencing the prediction results. Then we introduced a Word2Vec-based event representation learning approach, which provides high-quality input for the prediction model. Additionally, a transfer training iteration strategy based on different trace prefix lengths is employed to adapt the model to execution time differences across process variants. Experimental evaluations on benchmark datasets demonstrate that the proposed approach outperforms existing approaches in terms of both prediction accuracy and computational efficiency, confirming its effectiveness and robustness for predictive process monitoring.

This paper mainly focuses on activity and temporal features for remaining time prediction. Future research will explore the integration of additional event attributes, such as resource involvement, cost information, and organizational context, to further improve model performance and generalization. Moreover, while the deep learning model achieves superior accuracy, its explainability remains limited. Enhancing the explainability and transparency of remaining time prediction models will therefore be an important and promising direction for our future work.

7- Declarations

7-1- Author Contributions

Conceptualization, C.L.; methodology, X.X.; software, X.X.; validation, T.L.; formal analysis, N.G.; investigation, T.L.; data curation, N.G.; writing—original draft preparation, N.G.; writing—review and editing, C.L.; visualization, N.G.; supervision, C.L.; project administration, Q.Z.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

7-2- Data Availability Statement

The data presented in this study are available on request from the corresponding author.

7-3- Funding

This work was partly supported by the National Natural Science Foundation of China (No. 62472264 and 52374221), Natural Science Distinguished Youth Foundation of Shandong Province under Grant ZR2025QA13, Education Ministry Humanities and Social Science Research Youth Fund Project of China under grant 24YJCZH194, and national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project - UID/04152/2025 - Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS and UID/PRR/04152/2025.

7-4- Institutional Review Board Statement

Not applicable.

7-5- Informed Consent Statement

Not applicable.

7-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

8- References

- [1] Teinemaa, I., Dumas, M., Rosa, M. La, & Maggi, F. M. (2019). Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Transactions on Knowledge Discovery from Data*, 13(2), 1–57. doi:10.1145/3301300.
- [2] Liu, C., Wang, Y., Wen, L., Cheng, J., Cheng, L., & Zeng, Q. (2024). Discovering Hierarchical Multi-Instance Business Processes from Event Logs. *IEEE Transactions on Services Computing*, 17(1), 142–155. doi:10.1109/TSC.2023.3335360.
- [3] Vera-Baquero, A., Colomo-Palacios, R., Molloy, O., & Elbattah, M. (2015). Business process improvement by means of Big Data based Decision Support Systems: a case study on Call Centers. *International Journal of Information Systems and Project Management*, 3(1), 5-26.
- [4] Tax, N., Verenich, I., La Rosa, M., & Dumas, M. (2017). Predictive business process monitoring with LSTM neural networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10253 LNCS, 477–492. doi:10.1007/978-3-319-59536-8_30.
- [5] Van Der Aalst, W. M. P., Schonenberg, M. H., & Song, M. (2011). Time prediction based on process mining. *Information Systems*, 36(2), 450–475. doi:10.1016/j.is.2010.09.001.
- [6] Liu, C. (2022). Formal Modeling and Discovery of Multi-instance Business Processes: A Cloud Resource Management Case Study. *IEEE/CAA Journal of Automatica Sinica*, 9(12), 2151–2160. doi:10.1109/JAS.2022.106109.
- [7] Zhao, H. Y., Li, S. B., Chen, Q. K., & Cao, J. (2019). Method of time prediction for business process. *Journal of Chinese Computer Systems*, 40(2), 280-286.
- [8] Rogge-Solti, A., & Weske, M. (2015). Prediction of business process durations using non-Markovian stochastic Petri nets. *Information Systems*, 54, 1–14. doi:10.1016/j.is.2015.04.004.
- [9] Navarin, N., Vincenzi, B., Polato, M., & Sperduti, A. (2017). LSTM networks for data-aware remaining time prediction of business process instances. *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings*, 2018-January, 1–7. doi:10.1109/SSCI.2017.8285184.
- [10] Bukhsh, Z. A., Saeed, A., & Dijkman, R. M. (2021). Processtransformer: Predictive business process monitoring with transformer network. *arXiv Preprint, arXiv:2104.00721*. doi:10.48550/arXiv.2104.00721.
- [11] Wang, J., Yu, D., Liu, C., & Sun, X. (2019). Outcome-oriented predictive process monitoring with attention-based bidirectional LSTM neural networks. *Proceedings - 2019 IEEE International Conference on Web Services, ICWS 2019 - Part of the 2019 IEEE World Congress on Services*, 360–367. doi:10.1109/ICWS.2019.00065.
- [12] Guo, N., Liu, C., Mo, Q., Cao, J., Ouyang, C., Lu, X., & Zeng, Q. (2025). Business Process Remaining Time Prediction Based on Incremental Event Logs. *IEEE Transactions on Services Computing*, 18(3), 1308–1320. doi:10.1109/TSC.2025.3562338.
- [13] Weinzierl, S., Zilker, S., Dunzer, S., & Matzner, M. (2024). Machine learning in business process management: A systematic literature review. *Expert Systems with Applications*, 253, 124181. doi:10.1016/j.eswa.2024.124181.
- [14] Folino, F., Guarascio, M., & Pontieri, L. (2014). Mining predictive process models out of low-level multidimensional logs. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8484 LNCS, 533–547. doi:10.1007/978-3-319-07881-6_36.
- [15] Leontjeva, A., Conforti, R., Di Francescomarino, C., Dumas, M., & Maggi, F. M. (2015). Complex symbolic sequence encodings for predictive monitoring of business processes. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9253, 297–313. doi:10.1007/978-3-319-23063-4_21.
- [16] Polato, M., Sperduti, A., Burattin, A., & Leoni, M. de. (2018). Time and activity sequence prediction of business process instances. *Computing*, 100(9), 1005–1031. doi:10.1007/s00607-018-0593-x.
- [17] Verenich, I., Nguyen, H., Rosa, M. La, & Dumas, M. (2017). White-box prediction of process performance indicators via flow analysis. *ACM International Conference Proceeding Series, Part F128767*, 85–94. doi:10.1145/3084100.3084110.
- [18] Cao, R., Zeng, Q., Ni, W., Duan, H., Liu, C., Lu, F., & Zhao, Z. (2023). Business process remaining time prediction using explainable reachability graph from gated RNNs. *Applied Intelligence*, 53(11), 13178–13191. doi:10.1007/s10489-022-04192-x.
- [19] Aalikhani, R., Fathian, M., & Rasouli, M. R. (2024). Comparative Analysis of Classification-Based and Regression-Based Predictive Process Monitoring Models for Accurate and Time-Efficient Remaining Time Prediction. *IEEE Access*, 12, 67063–67093. doi:10.1109/ACCESS.2024.3397185.

- [20] Cheng, L., Du, L., Liu, C., Hu, Y., Fang, F., & Ward, T. (2024). Multi-modal fusion for business process prediction in call center scenarios. *Information Fusion*, 108, 102362. doi:10.1016/j.inffus.2024.102362.
- [21] Evermann, J., Rehse, J. R., & Fettke, P. (2017). A deep learning approach for predicting process behaviour at runtime. *Lecture Notes in Business Information Processing*, 281, 327–338. doi:10.1007/978-3-319-58457-7_24.
- [22] Nguyen, A., Chatterjee, S., Weinzierl, S., Schwinn, L., Matzner, M., & Eskofier, B. (2021). Time Matters: Time-Aware LSTMs for Predictive Business Process Monitoring. *Lecture Notes in Business Information Processing*, 406 LNBIP, 112–123. doi:10.1007/978-3-030-72693-5_9.
- [23] Ni, W., Sun, Y., Liu, T., Zeng, Q., & Liu, C. (2020). Business process remaining time prediction using bidirectional recurrent neural networks with attention. *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, CIMS*, 26(6), 1564–1572. doi:10.13196/j.cims.2020.06.013.
- [24] Khan, A., Le, H., Do, K., Tran, T., Ghose, A., Dam, H., & Sindhgatta, R. (2021). DeepProcess: Supporting Business Process Execution Using a MANN-Based Recommender System. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 13121 LNCS, 19–33. doi:10.1007/978-3-030-91431-8_2.
- [25] Pasquadibisceglie, V., Appice, A., Castellano, G., & Malerba, D. (2019). Using convolutional neural networks for predictive process analytics. *Proceedings - 2019 International Conference on Process Mining, ICPM 2019*, 129–136. doi:10.1109/ICPM.2019.00028.
- [26] Taymouri, F., Rosa, M. La, Erfani, S., Bozorgi, Z. D., & Verenich, I. (2020). Predictive business process monitoring via generative adversarial nets: The case of next event prediction. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12168 LNCS, 237–256. doi:10.1007/978-3-030-58666-9_14.
- [27] Amiri Elyasi, K., van der Aa, H., & Stuckenschmidt, H. (2024). PGTNet: A Process Graph Transformer Network for Remaining Time Prediction of Business Process Instances. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 14663 LNCS, 124–140. doi:10.1007/978-3-031-61057-8_8.
- [28] Liu, C., Liu, W., Guo, N., Song, R., Gu, Y., Cheng, L., & Zeng, Q. (2025). Comparative evaluation of encoding techniques for workflow process remaining time prediction for cloud applications. *Journal of Cloud Computing*, 14(1), 1–12. doi:10.1186/s13677-025-00763-8.
- [29] Guo, N., Liu, C., Li, C., Zeng, Q., Ouyang, C., Liu, Q., & Lu, X. (2024). Explainable and Effective Process Remaining Time Prediction Using Feature-Informed Cascade Prediction Model. *IEEE Transactions on Services Computing*, 17(3), 949–962. doi:10.1109/TSC.2024.3353817.
- [30] Church, K. W. (2017). Word2Vec. *Natural Language Engineering*, 23(1), 155–162. doi:10.1017/S1351324916000334.
- [31] Bradbury, J., Merity, S., Xiong, C., & Socher, R. (2017). Quasi-Recurrent Neural Networks. *International Conference on Learning Representations, ICLR 2017*, 1–11.
- [32] Fu, Y., Liang, Z., & You, S. (2021). Bidirectional 3D Quasi-Recurrent Neural Network for Hyperspectral Image Super-Resolution. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 2674–2688. doi:10.1109/JSTARS.2021.3057936.
- [33] Cao, R., Ni, W., Zeng, Q., Lu, F., Liu, C., & Duan, H. (2021). Remaining time prediction for business processes with concurrency based on log representation. *China Communications*, 18(11), 76–91. doi:10.23919/JCC.2021.11.006.
- [34] van Dongen, B. (2012): BPI Challenge 2012. Version 1. 4TU.ResearchData. dataset. doi:10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f.
- [35] Verenich, I. (2016). Helpdesk. Mendeley Data: Elsevier, Amsterdam, Netherlands. Available online: <https://mendeley.figshare.com/articles/dataset/Helpdesk/8951084> (accessed on March 2026).
- [36] Mannhardt, F. (2017): Hospital Billing - Event Log. Version 1. 4TU.ResearchData. dataset. doi:10.4121/uuid:76c46b83-c930-4798-a1c9-4be94dfb741.