Computing and Informatics, Vol. 44, 2025, 828–852, doi: 10.31577/cai_2025_4_828

ENSEMBLE PREDICTION OF BUSINESS PROCESS REMAINING TIME BASED ON RANDOM FOREST AND XGBOOST

Yinhua Tian*

College of Intelligent Equipment
Shandong University of Science and Technology
Taian 271000, China
&
College of Computer
Shandong Xiehe University
Jinan 250109, China
e-mail: skdxxtyh@163.com

Yan Su

School of Artificial Intelligence
Shandong Vocational University of Foreign Affairs
Weihai 264504, China
e-mail: sy19972023@163.com

Ruizhe Zhang

College of Intelligent Equipment
Shandong University of Science and Technology
Taian 271000, China
e-mail: zhangrz20010616@163.com

Yuyue Du, Nana Zhou, Xueqiang Gao

College of Computer
Shandong Xiehe University
Jinan 250109, China
e-mail: yydu001@163.com, nanazhou86@gmail.com, beyond7691@163.com

Abstract. The business processes in the information system are complex and diverse, and a single machine learning method often relies excessively on the noise or specific patterns in the training data. When dealing with large datasets, the calculation amount of the model is heavy, resulting in poor performance on new data, and it is difficult to achieve accurate monitoring and prediction of business processes. For this reason, a two-layer machine learning framework is presented using stacking technology – Serial Stacking Framework. Based on the event log, the method carries out random grouping sampling with placement, trains the multi-objective regression model, and applies multiple machine learning models to predict in series. Generally speaking, it is to use the prediction results of the previous model to generate training data and use it for the prediction of the latter model, in order to achieve the sequential accumulation of the prediction efficiency of multiple models. Random Forest and XGBoost are used as specific stack ensemble models for prediction, and the proposed method is evaluated against the existing advanced method through experiments. The results show that the average absolute error of the model built by the serial stacking framework with random group sampling and multi-objective regression is at least 2.14 % lower than that of the single machine learning model, the conventional stacking frameworks and the latest methods.

Keywords: Business processes, remaining time prediction, stacking ensemble, random forest, XGBoost

1 INTRODUCTION

Nowadays, Business Process Management (BPM) is widely used, especially the rapid development of big data has established the basis for advancing business process management [1, 2]. Business process is to standardize, optimize and automate the workflow of an enterprise or organization to achieve efficient, smooth and sustainable development of work. Process mining is a branch of data mining that aims to reveal insights from process-related data, track and enhance business processes. By analyzing past event logs, business process mining technology can extract valuable knowledge.

Recently, the research on Predictive Process Monitoring (PPM) technology has gradually attracted the attention of scholars at home and abroad [3], and has become a crucial aspect of business process mining. PPM addresses enterprises' needs to predict specific future moments or states.

PPM builds a prediction model by analyzing the historical execution logs to predict several quantifiable targets of the current process. These indexes include the remaining time for execution [4], the upcoming activities to perform and their potential execution times [5], the final process execution result [6], resource utilization and quality indicators. The objective of predictive business process monitoring

* Corresponding author

is to judge whether the cases in the current process have violations, timeouts or exceptions through real-time monitoring and analysis of the operation of the prediction model. By estimating the remaining time for business processes, enterprises can better plan and schedule resources to optimize execution efficiency of business processes. It can also solve potential problems in advance, improve ability of enterprises to cope with risks and survive, thus promote sustainable development and innovation of enterprises [7].

The study of remaining time prediction in PPM is mainly discussed in this paper. The prediction of remaining time is not only beneficial to the better operation of enterprises, but also has an impact on personal life. For example, the waiting time for banking business processing and the waiting time for medical treatment are predicted. Accurately predicting remaining time allows individuals to better organize their schedules. By understanding the time required to complete specific tasks or projects, individuals can better organize their schedules, thereby minimizing delays and reducing the likelihood of unexpected emergencies.

Traditional business process remaining time prediction is mainly realized by analyzing static business process models and historical logs. These methods usually predict the remaining time based on the static modeling of business processes. For example, researchers use a prediction method of absolute remaining time based on prefix trace representation learning method and attention mechanism [8], stochastic Petri nets [9] to outline the structure and workflow of business processes. In contemporary periods, many researchers have begun to apply machine learning technology and even deep learning technology for the task of predicting the remaining time, and obtained excellent prediction results. But there are still many problems that have not been effectively solved. The specific problems are as follows:

1. There are some challenges in sequence data processing, such as limited modeling ability, which limits its performance in terms of prediction accuracy, and there is still room for improvement.
2. Predicting the correlation between the data of remaining time for different tasks in the business process, and how to effectively predict the remaining time.
3. The singularity of the model is reflected in its relatively fixed structure, which is difficult to flexibly adapt to different types of data and task requirements. This oneness limits its applicability and performance when dealing with complex data and diversified tasks, so it is necessary to further explore more flexible and diversified model structures and methods.

In view of the above problems, random grouping sampling of data is conducted, and a multi-objective regression model is constructed, which solves the correlation problem between serial data processing and data. The above processing work is applied to the two-layer machine learning framework proposed in this paper, namely the Serial Stacking Framework (SSF), and comparative experiments are performed on five real datasets. The results indicate that the accuracy of the proposed method is considerably enhanced compared to other individual methods. SSF is shown

in Figure 1. The specific process is as follows: First, preprocessing and feature engineering the event log, and then using the data obtained to train the models respectively to obtain the prediction models of the models. Inputting the data into the two models to obtain the first prediction results, and then merging the obtained prediction results with data, and using them to train the new Random Forest and XGBoost to obtain the new Random Forest model and XGBoost prediction model. After data preprocessing and feature engineering, the instance being executed is directly input into RF-XGB and XGB-RF models, and finally the prediction results are obtained.

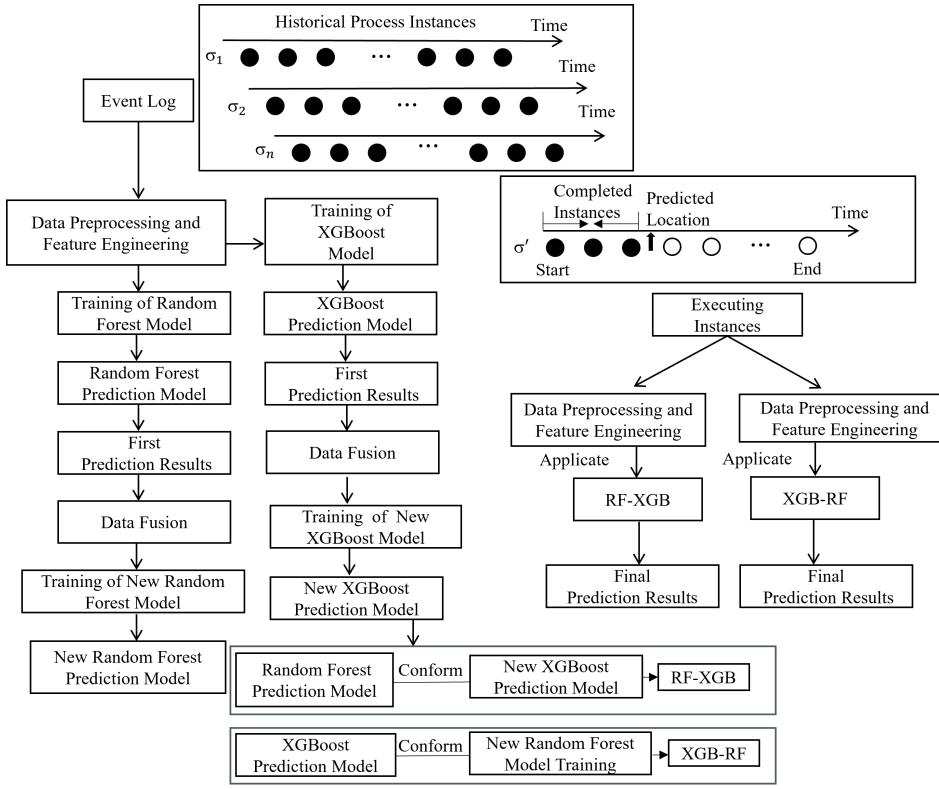

Figure 1. Serial stacking framework

The main contributions of this paper are as follows:

1. Randomly sample the data and divide it according to label groups, so that each group contains the same category label. This enhances the model's computational efficiency and lowers its complexity.
2. Building multiple objective regression. It combines multiple related regression

tasks into a whole model, and predicts multiple objective variables at the same time. This approach can reduce the number of models and enhance their overall performance, make better use of the correlation between targets, and improve the accuracy and efficiency of prediction.

3. For predicting the remaining time in business processes, the SSF prediction framework is proposed. The serial prediction of each basic model means that SSF can combine the advantages of multiple basic models and enhance the model's prediction accuracy and generalization capability by utilizing multi-level feature abstraction and integration. Because SSF combines the prediction results of multiple models, it has strong influence on noise and abnormal data, thus improving the robustness of the model.

2 RELATED WORK

At present, predicting the remaining time for business processes has emerged as a key focus in process mining, which has important research and practical value. It can help enterprises improve their control over business processes and optimize resource allocation. This leads to increased efficiency, lower costs, and enhanced customer satisfaction. Research on estimating the remaining time for business processes can typically be divided into three main approaches.

2.1 Prediction of the Remaining Time in Business Processes Based on the Process Model

The challenge of predicting remaining time was first addressed by van der Aalst et al. [10] in 2005, who proposed a method based on transition systems. To estimate the remaining time, this approach uses the constructed multi-level abstract transformation system to trace the state changes of all process instances that may occur in the event log. While recording the corresponding time information for each state. In addition, Rogge-Solti and Weske [11] introduced a method for simulating running process instances in their research, and estimated the remaining time by mining the stochastic Petri nets of event logs. Specifically, the data in the event log is specialized to construct a Petri net model, and simulations are conducted using Petri nets with random attributes. Simulating the execution of the process instance, and estimating the remaining time based on the simulation outcomes. However, the quality and accuracy of the process models are affected by the historical data used. If the historical data are inaccurate or incomplete, the performance of models may be affected, and it is difficult to effectively consider the impact of external factors on business processes.

2.2 Estimation of Remaining Time in Business Processes Through Machine Learning Methods

With the ongoing advancements in machine learning and deep learning technologies, many researchers begin to apply these technologies for forecasting the remaining time in business processes. Folino et al. [12] proposed a clustering based approach for predicting remaining time that used logical rules to represent the clustering model. Polato et al. [13] introduced a data-driven state transition system, which employed naive Bayesian classification to model existing state nodes and support vector machines to predict the remaining time. Bevacqua et al. [14] employed clustering and regression techniques to develop models for predicting remaining time in process instances across various variants. Ni et al. further improved the precision of predicting the remaining time of the business process through introducing the attention mechanism [15]. Xu et al. proposed a method to estimate the remaining time of the business process using a bidirectional recurrent neural network combined with an attention mechanism, which solved and improved the quantity variance and correlation between different length trace prefixes in process instances [16]. However, these methods generally rely on historical data during training, and business processes may change over time or in different environments. As a result, the models may fail to generalize well to new business process data, potentially leading to a decline in predictive performance. Chen et al. further introduced a multi-task prediction method for transfer learning [17]. But this method usually learns from historical data during training, and business processes can evolve over time and in various environments.

2.3 Forecasting Remaining Time in Business Processes Using Deep Learning

In recent years, several researchers have successfully integrated deep neural networks with various analytical techniques. Nguyen et al. proposed using cost sensitive learning on the basis of LSTM, which initially solved the problem of class imbalance and improved prediction accuracy [18], but reduced the interpretability of the model. Wahid et al. proposed a parallel structured model, which includes convolutional neural networks and multilayer perceptron combined with embedding layers, for predicting remaining time in the medical field [19]. However, this approach increased the computational resource demands. Bukhsh et al. introduced Transformer to solve long sequence problems and constructed a Process Transformer model [20]. However, the modeling of location information is not precise enough, and the model may not achieve the optimal results when dealing with tasks that require precise location information. Cao et al. [21] proposed constructing Petri nets and their reachable graphs to map RNN hidden states, to enhance prediction accuracy, but enhanced the complexity of the model. Huang et al. [22] introduced a predictive business process monitoring method founded on concept drift, which solves the problem of poor real-time prediction, but the model has a strong dependence on data.

In conclusion, a stackable integration framework SSF based on Random Forest and XGBoost for random grouping sampling and multi-objective regression [23] is proposed to improve the performance of the generalization model and efficiently and accurately predict the remaining duration of business processes.

The reason why Random Forest and XGBoost are selected as the basic model is that Random Forest and XGBoost are ensemble learning methods, but they differ in treatment methods and intensity. Initially, Random Forest is used for feature selection and model training to achieve relative stable baseline model. Later, XGBoost is used to further optimize and improve the output of Random Forest, so as to obtain higher prediction accuracy. This combination Random Forest not only retains the generalization ability of Random Forest also makes full use of XGBoost's precision improvement advantages.

3 BASIC CONCEPTS

This section primarily covers the fundamental concepts of predicting remaining time in business processes, including events, trace, trace prefix, and more.

Definition 1 (Events)**.** In business process mining, events refer to identifiable, time related business activities or state changes, usually in the form of records in the event log data, employed to analyze and model the behavior and characteristics of the process. $e = (a, id, T_{start}, T_{end}, p_1, \ldots, p_m)$ is a multivariate group used to represent events. The execution activity of an event is represented by a. The id identifies the process instance associated with the event. T_{start} represents the start time of event execution. The end time of event execution is displayed with T_{end}. The attributes of the event are expressed as p_1, p_2, …, p_m, including event ID, event type, event status, etc. Table 1 is a partial portion of an event log case.

From the perspective of process mining, online shopping process can be decomposed into a series of events. For example, in a shopping website, "add shopping cart", "generate order" and "payment" can be regarded as events. These events contain specific attributes and timestamps, which can be recorded and used as training data to establish business process models. By analyzing these event sequences, people can build a business process model which describes the shopping process, and subsequently use this model to estimate and analyze the remaining time, so as to understand the remaining time estimation in each stage of the shopping process. This method helps to improve the efficiency and user experience of shopping websites.

Definition 2 (Trace)**.** In business process mining, a denotes the sequence generated by executing an instance. It consists of a non-empty, finite sequence of all events in the case, organized in chronological order, which can be represented as $\sigma = \left(e_1, \ldots, e_{|\sigma|}\right)$. Wherein, for $\forall\ 1 \leq i \leq |\sigma|, \quad e_i$ represents the i^{th} executed event, and $|\sigma|$ indicates the total number of events within the trace. The event occurs only once.

Definition 3 (Trace Prefix)**.** The trace prefix denotes a partial sequence of trace, usually including a series of events that have occurred, and is used to predict and analyze real-time and future events in business processes. It can be used $\sigma^{(k)} = (e_1, \ldots, e_k)$, where, $1 \leq k \leq |\sigma|$. The remaining time of the trace prefix is $RT(\sigma, k) = e_{|\sigma|} \cdot T_{end} - e_k \cdot T_{end}$.

Definition 4 (Process Instances)**.** A process instance refers to an independent execution process generated according to a predefined template when executing a specific business process. The tuple is represented as $c = (Cid, \sigma, Z_1, \ldots Z_n)$. Cid represents the identification of the process instance, while σ indicates the path included in the process instance. $Z_1, \ldots, Z_m$ indicates the attributes of the process instance.

Definition 5 (Event Logs)**.** The event log is a database file that records various activities of the system, and details the execution of each step in one or more processes. Event logs record the historical execution of business processes, and contain information about all activity instances. It can be said that each activity instance corresponds to a record in the event log, and the entire event log is a collection of all process instances. It can be expressed by $L = \{c_1, \ldots, c_l\}$. Where, l indicates the number of processes instances c included in the event log L.

Case id	**Event id**	**Timestamp**	**Activity**	**Resource**	**Cost**
1	35654423	30-12-2010:11.02	register request	Pete	50
	35 654 424	31-12-2010:10.06	examine thoroughly	Sue	400
	35 654 425	05-01-2011:15.12	check ticket	Mike	100
	35 654 426	06-01-2011:11.18	decide	Sara	200
	35 654 427	07-01-2011:14.24	reject request	Pete	200
2	35654483	30-12-2010:11.32	register request	Mike	50
	35 654 485	30-12-2010:12.12	check ticket	Mike	100
	35 654 487	30-12-2010:14.16	examine thoroughly	Pete	400
	35 654 488	05-01-2011:11.22	decide	Sara	200
	35 654 489	08-01-2011:12.05	pay compensation	Ellen	200
...	...	...	...	...	...

Table 1. Event log case fragments

4 METHOD INTRODUCTION

The remaining time prediction framework of the business process based on two-layer machine learning proposed in this paper is shown in Figure 2. This section will introduce in detail the random grouping sampling and multi-objective regression method proposed in this paper, as well as the business process remaining time prediction framework SSF based on two-layer machine learning.

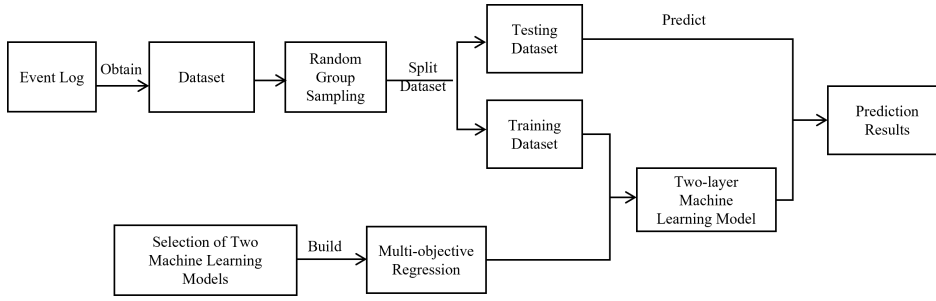

Figure 2. Remaining time prediction for business processes using a two-layer machine learning approach

4.1 Category Balanced Bootstrap Strategy

After determining the Random Forest and XGBoost, the correlation between decision trees plays an important role in the error of the original model. In practical applications, due to the imbalance of training data or the characteristics of sample distribution, some decision trees will have high correlation. This correlation will influence the accuracy of model prediction.

To address this issue, a method called category balanced bootstrap strategy is presented in this paper. This strategy aims to ensure that each category has enough samples in the new dataset by resampling training data. Multiple different balanced datasets can be generated by down sampling most classes and up sampling a few classes. Consequently, each decision tree is exposed to a unique data distribution during training. This heterogeneity among the training datasets increases the diversity among the individual trees, thereby reducing correlation and ultimately improving the precision of the model's predictions. The main steps of the category balanced bootstrap strategy are as follows:

1. **Preparating data.** First, obtaining the original dataset $D = \{(X_i, y_i)\}_{i=1}^{N}$, where X_i represents the feature set of the i^{th} sample, and y_i represents the target variable (category label) of the i^{th} sample. The processed dataset is D' to ensure that the category label of each sample in the dataset is clear.
2. **Separating categories.** According to the category of the target variable, the majority and minority samples in the dataset are separated. $y_i = 0$ indicates the majority class, and $y_i = 1$ indicates the minority class. The dataset D' is divided into the majority sample set D_{maj} and the minority sample set D_{min}. The specific formula is shown in Equations (1) and (2).

$$D_{maj} = \{(X_i, y_i) \mid y_i = 0\}, \tag{1}$$

$$D_{min} = \{(X_i, y_i) \mid y_i = 1\}. \tag{2}$$

The separation of categories adopts two classification strategies, namely, down sampling majority categories and up sampling minority categories. The down sampling of most classes is to randomly sample from the majority class sample set, so that the number of samples is the same as that of the minority class or reduced to the required proportion. The number of minority samples is n_{min}, and the majority sample set after down sampling is shown in Equation (3).

$$D_{maj}^{down} = RandomSample\left(D_{maj}, n_{min}\right). \tag{3}$$

The upsampling of minority class refers to the random repeated sampling from the minority class sample set, so that the number of samples is the same as that of the majority class or increased to the required proportion. The number of samples for most classes is n_{maj}, and the minority sample set after upsampling is shown in Equation (4).

$$D_{min}^{up} = RandomSample\left(D_{min}, n_{maj}\right). \tag{4}$$

3. Consolidating data. In step (2), the majority sample D_{maj}^{down} after down sampling is merged with the minority sample D_{min}^{up} after up sampling to form a new balanced dataset $D_{balanced}$, as shown in Equation (5).

$$D_{balanced} = D_{maj}^{down} \cup D_{min}^{up}. \tag{5}$$

4. Reseparating sampling. Repeat the above sampling steps for k times to generate k balanced datasets.

5. Separating feature and target variables. Reseparating characteristics and target variables mean separating characteristics and target variables from the balanced dataset. See Equation (6) for details.

$$D_{balanced}^{i} = \left\{\left(X_j^i, y_j^i\right)\right\}_{i=1}^{N_i}. \tag{6}$$

The characteristic matrix is $X^i = \left[X_1^i, X_2^i, \ldots, X_{N_i}^i\right]$, and the target vector is $y^i = \left[y_1^i, y_2^i, \ldots y_{N_i}^i\right]$.

By balancing the dataset, the decision trees can avoid over fitting the patterns of majority classes, and promote the model to learn more diverse features and patterns. This diversity will also reduce the correlation between decision trees. The specific methodology of the category balanced bootstrap strategy is depicted in Figure 3.

4.2 Construction of Multi-Objective Regression

Predicting the remaining time for business processes usually involves multiple related target variables, such as the leftover execution time for different tasks. Through the creation of multi-objective regression, these related target variables can be predicted

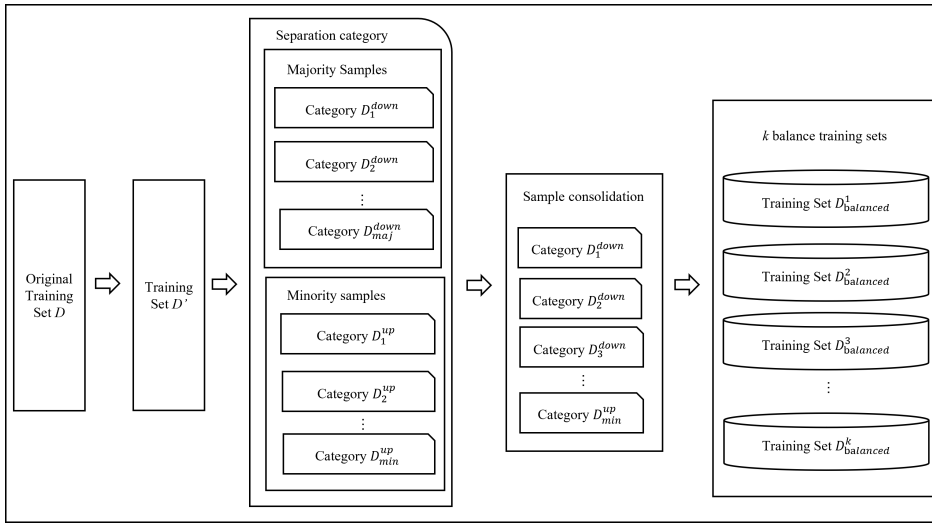

Figure 3. Category balanced bootstrap strategy process diagram

at the same time, rather than building multiple independent models for forecasting. The remaining time for execution of different tasks may be related or dependent. As an illustration, the execution duration of some tasks may be affected by the execution time of the previous task. Constructing a multi-objective regression chain allows for the explicit consideration of these interdependencies, potentially enhancing prediction accuracy and stability. On the basis of single objective modeling, directly build a regression, set the variable e_n, and the output value is shown in Equation (7).

$$e_n = \{e_{n,1}, e_{n,2}, \cdots, e_{n,m}\}, \tag{7}$$

where m represents the output quantity.

The construction of a multi-objective regression chain is based on the modification of Equation (7), which involves placing the last predicted result e_m in Equation (7) back into the original dataset, as shown in Equation (8).

$$l_n \cup e_{n,1:m} = \left(l_n^{(1)}, l_n^{(2)}, \ldots, l_n^{(z)}, e_{n,1}, \ldots, e_{n,m}\right). \tag{8}$$

Equation (9) is obtained by further processing Equation (8), as shown in the following Equation.

$$e_n = \{e_{n,m+1}\}. \tag{9}$$

Based on the above equations, a complete multi-objective regression can be built to more accurately predict the remaining duration of the business process.

4.3 SSF Method

In view of the stability requirements of the model for forecasting the remaining time of business processes in practical applications, and in some application scenarios, the prediction results of a single regression model are usually not ideal. To address this, The SSF method for forecasting the remaining time of business processes is introduced in this paper. The SSF method comprises seven stages, detailed as follows. The training steps of two-layer Machine Learning are shown in Algorithm 1. While the process framework of the SSF is depicted in Figure 3. Seven stages of the SSF:

1. **Data preprocessing stage:** Dividing the original training dataset into two parts, namely training set T_1 and T_2.
2. **First layer model training stage:** The training set T_1 and T_2 are used to train Random Forest and XGBoost, respectively, and the best prediction models M_1^{RF} and M_1^{XGB} are obtained by grid search and cross validation.
3. **Preliminary data prediction stage:** Inputting the training set T_2 into the Random Forest prediction model M_1^{RF} to get the result set R_1; Feeding the training set T_1 into the XGBoost M_1^{XGB} to obtain the result set R_2.
4. **Data fusion stage:** the dataset for training T_2 is integrated with the result set R_1 to get the training set T_2', and the training set T_1 is integrated with the result set R_2 to get the training set T_1'.
5. **The second level model training stage:** The training set T_1' and training set T_2' are used to train Random Forest and XGBoost, respectively, and the best prediction models are obtained by grid search and cross validation to get M_2^{RF} and M_2^{XGB}.
6. **Building hybrid prediction model stage:** Integrating the models obtained in step 2) and step 5) to obtain the final hybrid prediction model.
7. **Model prediction stage:** Inputting the testing set into RF-XGB model and XGB-RF model to get prediction results $R_1\prime$ and R_2'.

In the SSF method, the model involves the training and prediction process of multiple models, so more computing resources are needed [24]. The computational complexity associated with the algorithm is determined by the time consumed for model training. The time complexity of training Random Forest is $O((k.a-k.min)*n*d*\log(n))$, where the data size of each cycle is variable, but in general, the cycle count is $O(k.a-k.min)$. Similarly, the complexity of XGBTrain is $O(n*d*\log(n))$, and the number of cycles is $O(k.max-k.a)$, so the total complexity is $O((k.max-k.a)*n*d*\log(n))$. Similarly, the complexity of XGBTrain is $O(n*d*\log(n))$, and the number of cycles is $O(k.max-k.a)$, so the total complexity is $O((k.max-k.a)*n*d*\log(n))$. The new Random Forest should be trained. The complexity of each training is the same as before, and the overall complexity

Algorithm 1 Training two-layer machine learning mode

Input: Training datasets $D = \{D_{k.min} \cup D_{k.min+1} \cup \cdots \cup D_{k.max}\}$.

Output: Two-layer machine learning model $\{M_{k.min} \cup M_{k.min+1} \cup \cdots \cup M_{k.max}\}$.

1: $a = (length(max - min - 1))/2; A[a-1] = \text{set}(\ );\ A[max - a - 1] = \text{set}(\ );$
// Define a, to initialize two empty sets
2: **for** i *in range*(a)
3: $A[a-1].add(D[i])$
// Traversing the previous a trace and putting it in the empty set
4: **for** i *in range*$(a$,*total_trace*$)$
5: $A[max - a - 1].add(D[i])$;
// Traversing the remaining traces and putting them in the empty set
6: **end for**
7: **end for**
8: **for** m in range$(k.min + 1, k.a)$
// Determining the number of training cycles of Random Forest
9: $M_1^{RF} \leftarrow RFTrain\left(D_{(k.min+1,\ldots k,\ldots a)}, M_{(k.min,\ldots k,\ldots a-1)}\right)$;
// Training Random Forest
10: **end for**
11: **for** n inrange$(k.a + 1, k.max)$;
// Determining the number of XGBoost training cycles
12: $M_1^{RF} \leftarrow XGBTrain\left(D_{(k.a+1,\ldots l,\ldots max)}, M_{(k.a,\ldots,\ldots,\ldots,max-1)}\right)$;
// Training XGBoost
13: **end for**
14: $M_1^{RF} \leftarrow [A_0, A_{a-1}]; M_1^{XGB} \leftarrow [A_a, A_{max-a+1}]$;
// Inputting the training sets obtained in step 3 and step 4
// into the Random Forest and XGBoost, respectively
15: $R_1 \leftarrow M_1^{RF}, R_2 \leftarrow M_1^{XGB}$; // Getting the output result of step 13
16: **for** i in range(p)
// Determining the number of training cycles for the new Random Forest
17: $M_2^{RF} \leftarrow N\_RFTrain\ \left(D_{(k.min+1,\ldots b,\ldots min+p)}, M_{(k.min,\ldots b,\ldots min+p-1)}\right)$;
// Determining a new Random Forest
18: **end for**
19: **for** i in range(q) // Determining the number of new XGBoost training cycles
20: $M_2^{XGB} \leftarrow N\_XGBTrain\ \left(D_{(k.a+1,\ldots c,\ldots min+q)}, M_{(k.a,\ldots c,\ldots min+q-1)}\right)$
// Training new XGBoost prediction model
21: **end for**
22: **return** $\left\{M_{XGB}^{RF} = M_1^{RF} + M_2^{XGB}; M_{RF}^{XGB} = M_1^{XGB} + M_2^{RF}\right\}$
// Outputting two-layer machine learning model

is $O\left(p * n * d * \log(n)\right)$. Similarly, the new XGBoost model should be trained. The overall complexity amounts to $O(q * n * d * \log(n))$.

Since the main time is spent on model training, and the complexity of these training steps is much higher than other steps, the overall complexity is mainly

determined by model training, that is, $O((k.a-k.min+k.max-k.a+p+q)*n*d*\log(n))$, which is simplified to $O((k.max-k.min+p+q)*n*d*\log(n))$. Here, n is the data amount of each round of training, and d is the feature dimension.

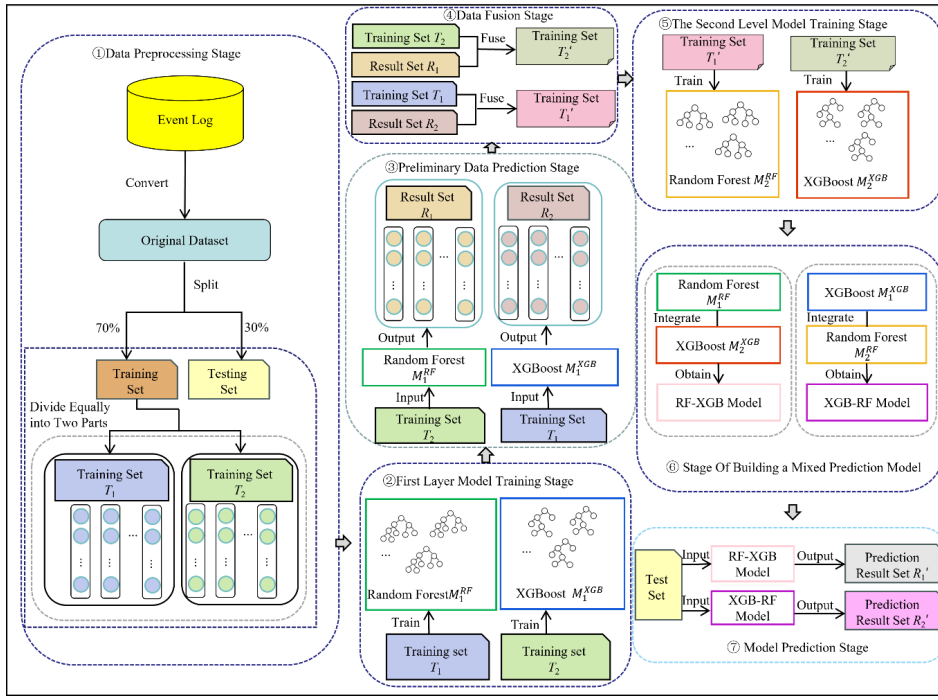

Figure 4. SSF process diagram

5 RELEVANT EXPERIMENTS AND ANALYSIS

This section explores and analyzes the prediction of remaining time in business processes method based on two-layer machine learning. First of all, it introduces the experimental setup, various datasets used and model evaluation indicators. Next, it evaluates the approach presented in this paper against other methods, analyzes the experimental effect of each model, and finally conducts ablation experimental analysis.

5.1 Experimental Setup

Python 2.7.3 was used for data processing and experimental comparisons in this study. The parameter settings of the model in the experiment are shown in Table 2.

When building a Random Forest in this experiment, the value range pertaining to decision trees is $[100, 200]$, the maximum depth range of trees is $\{3, 5, 7\}$, and the number range of feature selections is $\{\text{None}, \text{sqrt}, \log 2\}$. For XGBoost, the number of decision trees ranges from $[100, 200]$, tree depths are in $\{3, 5, 7\}$, the proportion of randomly selected features per tree ranges from $[0.8, 1.0]$, and the learning rate is within $\{0.01, 0.1, 0.5\}$.

Model	**Parameters**	**Parameter Range**
Random Forest	n_estimators	$[100, 200]$
	max_depth	$\{3, 5, 7\}$
	max_features	$\{\text{None}, \text{sqrt}, \log 2\}$
XGBoost	n_estimators	$[100, 200]$
	max_depth	$\{3, 5, 7\}$
	colsample_bytree	$[0.8, 1.0]$
	learning_rate	$\{0.01, 0.1, 0.5\}$

Table 2. Experimental parameter settings

5.2 Experimental Data

This study utilizes five event log datasets published by the 4TU. Centre for Research Data platform: Hospital_Billing, Helpdesk, BPIC_2012_A, BPIC_2012_W, and BPIC_2012_O. Table 3 displays the fundamental details of the five datasets mentioned above.

Dataset	Number of Events	Number of Activities	Maximum Trace Length	Minimum Trace Length
Hospital_Billing	451 359	18	217	1
Helpdesk	13 710	9	14	1
BPIC_2012_A	73 022	10	10	3
BPIC_2012_W	147 450	6	153	1
BPIC_2012_O	41 728	7	39	4

Table 3. Basic information of datasets

5.3 Evaluating Indicator

When predicting the remaining time pertaining to a business process, historical observations can be used as input to train a suitable model (such as Linear Regression, Random Forest, XGBoost, and others), and use this model to predict future remaining time. Mean absolute error (MAE) is used as a key measure to evaluate the prediction model.

For a set of traces with a given prefix trace length, it is assumed that there are historical observations y_1, y_2, …, y_t and corresponding predicted values $\hat{y}_1$, $\hat{y}_2$, …, $\hat{y}_t$. Then the MAE calculation formula of this group of traces is shown in Equation (10).

$$\mathrm{MAE}(f) = \sum_{\left(\sigma^{(k)},\mathrm{R}T(\sigma,k)\right)\in D)} \left| f\left(\sigma^{(k)} - \mathrm{R}T(\sigma,k)\right)\right|. \tag{10}$$

Among them, $\sigma^{(k)}$ stands for trace prefix, $f\left(\sigma^{(k)}\right)$ represents the estimated remaining time value of the trace prefix, and $RT(\sigma,k)$ represents the true value of the time remaining for the trace prefix.

In this experiment, the model's performance is assessed using 5-fold cross validation method. It partitions the dataset into five equal parts, and then conducts five training set and testing set. In each iteration, four segments serve as the training set, while one segment acts as the dataset, and different parts are selected as the testing set in turn. Ultimately, the mean of the five test results is used as The performance assessment metric of the model on the entire dataset.

5.4 Comparative Experiment

Comparing the methods proposed in this paper with traditional process model based methods, deep learning methods and the conventional stacking frameworks. The six benchmark methods selected include:

1. The method of predictive business process monitoring with an LSTM neural network introduced by Tax et al. [25], referred to as LSTM for short.
2. Time prediction technique utilizing migration system proposed by van der Aalst et al. [10]. During the experiment, the states of the migration system are abstractly represented as sets, multiple sets and sequences, named TS-set, TS-multi set and TS-sequence respectively.
3. Rogge-Solti and Weske [11] suggested the approach of using delayed Stochastic Petri nets for forecasting the remaining service execution duration, which is called SPN for short.
4. Ni and colleagues introduced a method utilizing an attention-based bidirectional recurrent neural network, referred to as Att-Bi-RNN [15].
5. Xu et al. proposed a technique utilizing a bidirectional quasi-cyclic neural network with attention mechanism, referred to as Trans-att-Bi-QRNN [16].
6. The conventional stacking method selected in this paper is the parallel stacking framework termed as RF||XGB.

MAE evaluation results of the methods and reference methods in this paper on different datasets are shown in Table 4, where RF-XGB and XGB-RF are the SSF method models proposed in this paper.

Comparison Method	Hospital_ Billing	Helpdesk	BPIC_ 2012_A	BPIC_ 2012_W	BPIC_ 2012_O
TS-set [10]	51.456	6.283	7.505	8.429	7.392
TS-multiset [10]	51.507	6.167	7.488	8.691	7.203
TS-sequence [10]	51.504	6.192	7.488	8.619	9.612
SPN [11]	78.018	6.337	8.880	8.516	6.385
LSTM [25]	42.050	3.542	3.588	8.021	7.993
Att-Bi-RNN [15]	32.187	3.299	3.438	5.821	5.863
Trans-Att-Bi-QRNN [16]	31.436	**2.423**	**2.373**	5.275	5.158
Decision Tree	31.906	3.156	3.422	5.912	5.235
Random Forest	30.899	3.432	3.366	5.177	5.130
XGBoost	31.019	3.453	3.400	5.212	**5.121**
RF||XGB	30.977	3.441	3.371	5.211	5.354
RF-XGB	30.781	3.435	3.332	**5.022**	5.165
XGB-RF	**30.763**	3.426	3.380	5.100	5.202

Table 4. Comparative experimental results of remaining time prediction methods

According to the results of comparative experiments, a conclusion is drawn that the two-layer machine learning framework using Random Forest and XGBoost has less error for forecasting the remaining duration of business processes than the techniques grounded in transition system and random Petri nets, LSTM, decision tree, single Random Forest, XGBoost and RF||XGB.

The MAE of the method in this paper is less than the result of using the depth learning method LSTM. On all datasets except the dataset from the Helpdesk MAE is superior to the Att-Bi-RNN method. On two of the five datasets, MAE is better than the Trans-Att-Bi-QRNN method, and MAE is less than the result of using the Decision Tree. The reason is that the method presented in this paper is more effective when dealing with large datasets, and the model can use more abundant data resources for training during the training process, thus significantly improving the training accuracy.

The Helpdesk, BPIC_2012_O and BPIC_2012_A contain significantly less data, with the number of events being only 13 710, 41 728, and 73 022, respectively. These figures are considerably smaller compared to the other two datasets. During the training process, the model was not fully optimized, and key information in the data was not fully utilized, which impacted the accuracy of the remaining time prediction. In contrast, the Hospital_Billing and BPIC_2012_W contain 451 359 and 147 450 events, respectively. The large scale of these datasets provides a wealth of training samples, enabling the model to identify more patterns and improve prediction accuracy. By fully leveraging these data, the model can better capture the complexity of business processes and enhance its generalization ability. Therefore, there is a noticeable gap between the prediction performance of this method on the Hospital_Billing and BPIC_2012_W datasets and its performance on the Helpdesk, BPIC_2012_O and BPIC_2012_A.

5.5 Analysis of Experimental Results on Segmentation Ratios of Different Datasets

The datasets such as Hospital_Billing are divided into training sets and testing sets, and the datasets are divided by three different segmentation ratios of 8:2, 7:3 and 5:5, respectively. The experimental prediction results MAE values obtained are shown in Figure 5, with the unit of days. R represents the segmentation ratio of each dataset in this paper. It can be seen from Figure 5 that the 7:3 method used in this paper is better for prediction when processing datasets. The 7:3 segmentation method utilizes 70 % of the data for model training and 30 % for performance evaluation. More data can be used for training. More data can help the model learn features and patterns more comprehensively, and larger testing sets can also be more representative of the entire data distribution to better assess the model's performance.

5.6 Ablation Experiment

To further illustrate the effectiveness of random grouping sampling on the dataset and the advantages of using multi-objective regression in the model, this section conducted ablation experiments on five datasets including Hospital_Billing for random grouping sampling and multi-objective regression. The results of the experiments are presented in Table 5.

Table 5 shows the experimental evaluation outcomes of the model with or without random grouping sampling and multi-objective regression processing. Obviously, models that undergo random group sampling or multi-objective regression processing alone have much better predictive performance than models that do not undergo both processing. Models that simultaneously undergo random group sampling and multi-objective regression processing perform the best in prediction. This indicates that random grouping sampling and multi-objective regression have a beneficial effect on the model's performance.

From Table 5, it can be seen that the prediction performance of random grouping sampling on the Helpdesk and BPIC_2012_W datasets is better than that of multi-objective regression processing, and the prediction performance of multi-objective regression processing is better on the other datasets. This indicates that random grouping sampling has a more significant predictive effect on large datasets, but both prediction processes improve the accuracy and reliability of model predictions.

Random grouping sampling reduces bias in the dataset, enabling the model to improve capture the overall features of the dataset, thereby improving prediction accuracy, especially for tasks that require high-precision prediction. This highlights the importance of adopting random grouping sampling in the data preprocessing stage. Compared to establishing multiple single objective regression models separately, multi-objective regression models can handle all target variables in one model, simplifying the modeling process and reducing the number and complexity

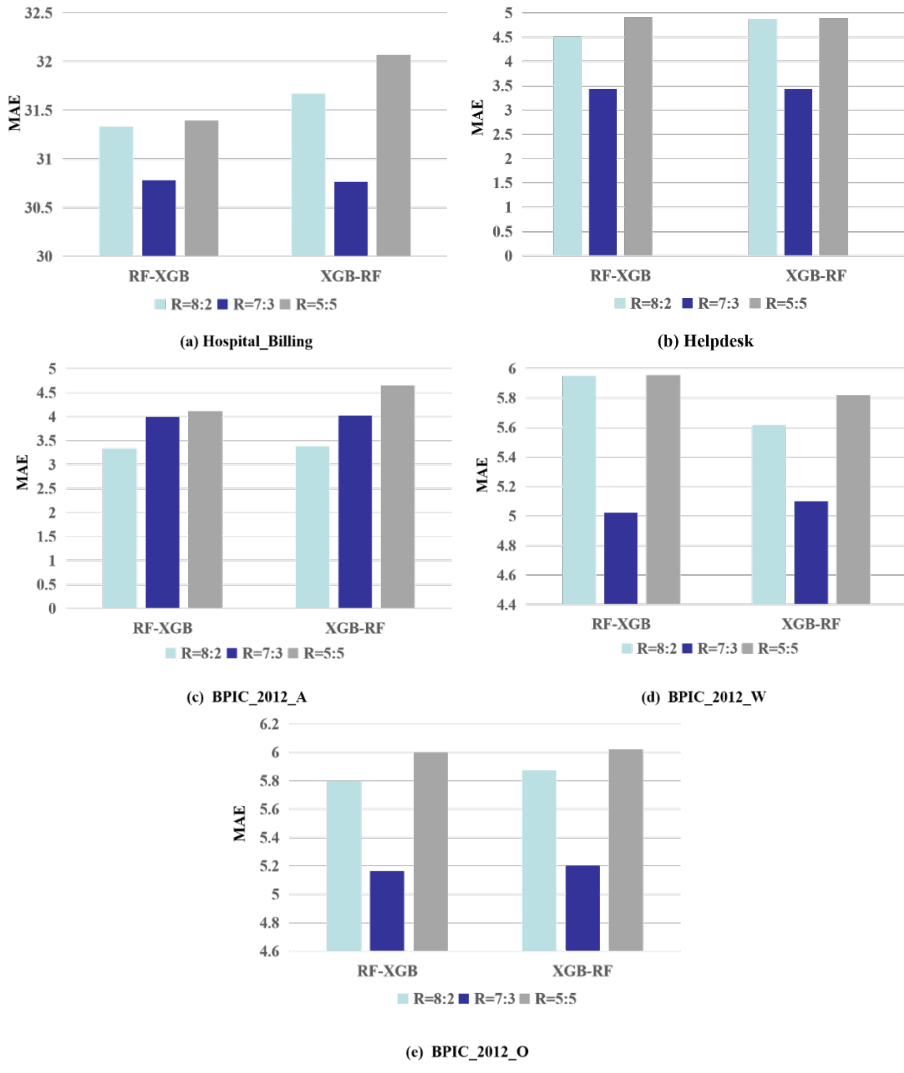

Figure 5. Comparison of MAE values of different segmentation ratios for each dataset

of models, thereby more accurately estimating the remaining duration of business processes.

Model	Preprocessing Methods	Hospital_ Billing	Help-desk	BPIC_ 2012_A	BPIC_ 2012_W	BPIC_ 2012_O
RF-XGB	Unprocessed	31.743	3.801	3.419	5.385	5.380
	Only Perform Random Group Sampling	31.061	3.659	3.421	5.104	5.311
	Only Perform Multi-objective Regression	31.325	3.550	3.356	5.216	5.262
	Perform Two Types of Processing	**30.781**	**3.435**	**3.332**	**5.022**	**5.165**
XGB-RF	Unprocessed	31.019	3.745	3.459	5.229	5.306
	Only Perform Random Group Sampling	30.844	3.694	3.417	5.146	5.299
	Only Perform Multi-objective Regression	31.154	3.629	3.396	5.322	5.256
	Perform Two Types of Processing	**30.763**	**3.426**	**3.380**	**5.100**	**5.202**

Table 5. Comparison of ablation test results

6 SUMMARY AND OUTLOOK

A technique for estimating the remaining time of business processes using two-layer machine learning is aimed to be studied in this paper. Taking two learning models, Random Forest and XGBoost, as examples, the SSF method is proposed, and its application in different fields will be extended to more business scenarios, such as healthcare, logistics, manufacturing, etc., which can effectively avoid errors caused by unknown execution times of certain process links. The time that users need to spend on their own actions is transparent, making it easier for them to proceed with the next task. And through the analysis of experimental results, it can be found that the SSF architecture performs well in predicting the remaining time of business processes, which can further improve the accuracy and stability of predictions.

However, the method proposed in this paper falls under the category of stacked ensemble learning, and the overall performance of this method is largely influenced by the selection and performance of the foundational model. If the performance of the basic model is poor, it could adversely affect the overall integrated model [26, 27]. Thus, exploring ways to enhance the model's overall performance, how to select the

basic model and evaluate its performance, these are also the next steps that need to be studied.

The future research prospects for estimating the remaining duration of business processes are as follows:

1. Deep learning models can be utilized for the SSF framework to solve the challenge of estimating the remaining time of business processes.
2. Enhance the interpretability of stacked ensemble learning models and utilize them to assess the remaining time of business processes.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 72101137, No. 61973180), the Education Ministry Humanities and Social Science Research Youth Fund Project of China (No. 21YJCZH150, No. 20YJCZH159), the Natural Science Foundation of Shandong Province (No. ZR2021MF117, No. ZR2022QF020), and the Key R & D Program (Soft Science) Project of Shandong Province (No. 2022RKY02009), the Shandong Digital Economy Research Base Project of Research Center of Shandong Province on "Xi Jinping Thought on Socialism with Chinese Characteristics for a New Era" and Shandong University of Science and Technology (No. SDSZJD202314).

REFERENCES

[1] Sarker, I. H.: Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, Vol. 2, 2021, No. 3, Art. No. 160, doi: 10.1007/s42979-021-00592-x.

[2] Meng, X.—Du, Z.: Research on the Big Data Fusion: Issues and Challenges. Journal of Computer Research and Development, Vol. 53, 2016, No. 2, pp. 231–246, doi: 10.7544/issn1000-1239.2016.20150874 (in Chinese).

[3] Marquez-Chamorro, A. E.—Resinas, M.—Ruiz-Cortés, A.: Predictive Monitoring of Business Processes: A Survey. IEEE Transactions on Services Computing, Vol. 11, 2018, No. 6, pp. 962–977, doi: 10.1109/TSC.2017.2772256.

[4] Neu, D. A.—Lahann, J.—Fettke, P.: A Systematic Literature Review on State-of-the-Art Deep Learning Methods for Process Prediction. Artificial Intelligence Review, Vol. 55, 2022, No. 2, pp. 801–827, doi: 10.1007/s10462-021-09960-8.

[5] Pasquadibisceglie, V.—Appice, A.—Castellano, G.—Malerba, D.: A Multi-View Deep Learning Approach for Predictive Business Process Monitoring. IEEE Transactions on Services Computing, Vol. 15, 2021, No. 4, pp. 2382–2395, doi: 10.1109/TSC.2021.3051771.

[6] Teinemaa, I.—Dumas, M.—La Rosa, M.—Maggi, F. M.: Outcome-Oriented Predictive Process Monitoring: Review and Benchmark. ACM Transactions on Knowledge Discovery from Data (TKDD), Vol. 13, 2019, No. 2, Art. No. 17, doi: 10.1145/3301300.

[7] van der Aalst, W.: Process Mining: Data Science in Action. Springer, 2016, doi: 10.1007/978-3-662-49851-4.

[8] Tian, Y.—Pang, X.—Yang, R.—Han, D.—Wang, L.—Du, Y.: Method for Predicting Absolute Remaining Time of Business Processes Based on Prefix Trace Representation Learning and Attention Mechanism. Computer Integrated Manufacturing Systems, Vol. 31, 2025, No. 5, pp. 1762–1778, doi: 10.13196/j.cims.2024.BPM12 (in Chinese).

[9] Rogge-Solti, A.—Weske, M.: Prediction of Business Process Duration Using Non-Markovian Stochastic Petri Nets. Information Systems, Vol. 54, 2015, pp. 1–14, doi: 10.1016/j.is.2015.04.004.

[10] van der Aalst, W. M. P.—Schonenberg, M. H.—Song, M.: Time Prediction Based on Process Mining. Information Systems, Vol. 36, 2011, No. 2, pp. 450–475, doi: 10.1016/j.is.2010.09.001.

[11] Rogge-Solti, A.—Weske, M.: Prediction of Remaining Service Execution Time Using Stochastic Petri Nets with Arbitrary Firing Delays. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (Eds.): Service Oriented Computing (ICSOC 2013). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 8274, 2013, pp. 389–403, doi: 10.1007/978-3-642-45005-1_27.

[12] Folino, F.—Guarascio, M.—Pontieri, L.: Mining Predictive Process Models Out of Low-Level Multidimensional Logs. In: Jarke, M. et al. (Eds.): Advanced Information Systems Engineering (CAiSE 2014). Springer, Cham, Lecture Notes in Computer Science, Vol. 8484, 2014, pp. 533–547, doi: 10.1007/978-3-319-07881-6_36.

[13] Polato, M.—Sperduti, A.—Burattin, A.—de Leoni, M.: Time and Activity Sequence Prediction of Business Process Instances. Computing, Vol. 100, 2018, No. 9, pp. 1005–1031, doi: 10.1007/s00607-018-0593-x.

[14] Bevacqua, A.—Carnuccio, M.—Folino, F.—Guarascio, M.—Pontieri, L.: A Data-Driven Prediction Framework for Analyzing and Monitoring Business Process Performances. In: Hammoudi, S., Cordeiro, J., Maciaszek, L. A., Filipe, J. (Eds.): Enterprise Information Systems (ICEIS 2013). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 190, 2013, pp. 100–117, doi: 10.1007/978-3-319-09492-2_7.

[15] Ni, W.—Sun, Y.—Liu, T.—Zeng, Q.—Liu, C.: Business Process Remaining Time Prediction Using Bidirectional Recurrent Neural Networks with Attention. Computer Integrated Manufacturing Systems, Vol. 26, 2020, No. 6, pp. 1564–1572, doi: 10.13196/j.cims.2020.06.013 (in Chinese).

[16] Xu, X. R.—Liu, C.—Li, T.—Guo, N.—Ren, C. G.: Business Process Remaining Time Prediction: An Approach Based on Bidirectional Quasi Recurrent Neural Network with Attention. Acta Electronica Sinica, Vol. 50, 2022, No. 8, pp. 1975–1984 (in Chinese).

[17] Chen, H.—Fang, X.—Fang, H.: Multi-Task Prediction Method of Business Process Based on BERT and Transfer Learning. Knowledge-Based Systems, Vol. 254, 2022, Art. No. 109603, doi: 10.1016/j.knosys.2022.109603.

[18] Nguyen, A.—Chatterjee, S.—Weinzierl, S.—Schwinn, L.—Matzner, M.—Eskofier, B.: Time Matters: Time-Aware LSTMs for Predictive

Business Process Monitoring. In: Leemans, S., Leopold, H. (Eds.): Process Mining Workshops (ICPM 2020). Springer, Cham, Lecture Notes in Business Information Processing, Vol. 406, 2021, pp. 112–123, doi: 10.1007/978-3-030-72693-5_9.

[19] Wahid, N. A.—Bae, H.—Adi, T. N.—Choi, Y.—Iskandar, Y. A.: Parallel-Structure Deep Learning for Prediction of Remaining Time of Process Instances. Applied Sciences, Vol. 11, 2021, No. 21, Art. No. 9848, doi: 10.3390/app11219848.

[20] Bukhsh, Z. A.—Saeed, A.—Dijkman, R. M.: ProcessTransformer: Predictive Business Process Monitoring with Transformer Network. CoRR, 2021, doi: 10.48550/arXiv.2104.00721.

[21] Cao, R.—Zeng, Q.—Ni, W.—Duan, H.—Liu, C.—Lu, F.—Zhao, Z.: Business Process Remaining Time Prediction Using Explainable Reachability Graph from Gated RNNs. Applied Intelligence, Vol. 53, 2023, No. 11, pp. 13178–13191, doi: 10.1007/s10489-022-04192-x.

[22] Huang, H.—Yang, Z.—Li, X.—Li, C.: Predictive Business Process Monitoring Method Based on Concept Drift. Journal of Computer Applications, Vol. 44, 2024, No. 10, pp. 3167–3176 (in Chinese).

[23] Wolpert, D. H.: Stacked Generalization. Neural Networks, Vol. 5, 1992, No. 2, pp. 241–259, doi: 10.1016/S0893-6080(05)80023-1.

[24] Liu, X.—Jing, L. P.—Yu, J.: Diverse and Authentic Task Generation Method for Robust Few-Shot Classification. Journal of Software, Vol. 35, 2024, No. 4, pp. 1587–1600, doi: 10.13328/j.cnki.jos.007014 (in Chinese).

[25] Tax, N.—Verenich, I.—La Rosa, M.—Dumas, M.: Predictive Business Process Monitoring with LSTM Neural Networks. In: Dubois, E., Pohl, K. (Eds.): Advanced Information Systems Engineering (CAiSE 2017). Springer, Cham, Lecture Notes in Computer Science, Vol. 10253, 2017, pp. 477–492, doi: 10.1007/978-3-319-59536-8_30.

[26] Wang, C.—Cao, J.: Interval-Based Remaining Time Prediction for Business Processes. In: Hacid, H., Kao, O., Mecella, M., Moha, N., Paik, H. Y. (Eds.): Service-Oriented Computing (ICSOC 2021). Springer, Cham, Lecture Notes in Computer Science, Vol. 13121, 2021, pp. 34–48, doi: 10.1007/978-3-030-91431-8_3.

[27] Li, M.—Ding, Z.: A Preface to the Special Issue on Emerging and Intelligent Information Services. Computing and Informatics, Vol. 39, 2020, No. 1–2, pp. 1–4, doi: 10.31577/cai_2020_1-2_1.

Yinhua Tian received her B.Sc. degree in computer science and technology, her M.Sc. degree and her Ph.D. degree in computer software and theory from the Shandong University of Science and Technology, Qingdao, China, in 2004, 2007 and 2018, respectively. She is currently an Associate Professor of computer science and technology with the Shandong University of Science and Technology. She has authored over 20 technical papers in journals and conference proceedings. Her current research interests include Petri nets, process mining, and optimization algorithms.

Yan Su received her B.Sc. degree from the Taishan College of Science and Technology, Shandong University of Science and Technology, in 2021. She is currently pursuing the M.S. degree with the College of Intelligent Equipment, Shandong University of Science and Technology, Taian, China. Her main research interests include predictive process monitoring and data mining.

Ruizhe Zhang received his B.Sc. degree from the Shandong University of Science and Technology, Jinan, China, in 2023. He is currently pursuing his M.Sc. degree with the College of Intelligent Equipment, Shandong University of Science and Technology, Taian, China. His main research interests include process mining and predictive process monitoring.

Yueyue Du received his B.Sc. degree from the Shandong University, Jinan, China, in 1982, his M.Sc. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1991, and his Ph.D. degree in computer application from the Tongji University, Shanghai, China, in 2003. He is currently Professor at the College of Computer, Shandong Xiehe University, Jinan, China, and Professor of computer science and technology at the Shandong University of Science and Technology, Qingdao, China. He has taken in over 10 projects supported by the National Nature Science Foundation, the National Key Basic Research Developing Program, and other important and key projects at provincial levels. He has published over 200 papers in domestic and international academic publications. His research interests are in formal engineering, Petri nets, real-time systems, Web services, and workflows.

Nana Zhou received her Ph.D. degree in mechanical engineering from the National Engineering Laboratory for Electric Vehicles, Beijing Institute of Technology, Beijing, China, in 2020, and completed her postdoctoral research at the Institute of Advanced Technology, Beijing Institute of Technology, in 2023. She is currently Associate Professor at the College of Computer, Shandong Xiehe University, Jinan, China. Her current research interests include state estimation, parameter identification, safety management, fault detection and precaution, and remaining life prediction of drive systems.

Xueqiang Gao received his Ph.D. degree in engineering from the Yantai Naval Aeronautical Engineering Institute, Yantai in 2009. He is currently Professor at the Institute of Computer Science, Shandong Xiehe University. His current research interests include signal and information processing, artificial intelligence, navigation and positioning technology, and visible light communication technology.