Ứng dụng Quá trình điểm thời gian trong giám sát quá trình kinh doanh

TÓM TẮT

Dự báo quá trình giám sát kinh doanh là một nhiệm vụ dạng chuỗi thời gian đầy thách thức do bản chất phức tạp và biến thiên của các quy trình kinh doanh, liên quan đến việc dự đoán các trường hợp đang diễn ra như hoạt động tiếp theo, hậu tố hoạt động và dự đoán thời gian còn lại trong một quy trình kinh doanh. Quá trình điểm thời gian được sử dụng rộng rãi để mô hình hóa chuỗi các sự kiện xảy ra ở các khoảng thời gian không đồng đều, để mô hình hóa thời gian xảy ra và nắm bắt các phụ thuộc thời gian giữa các sự kiện. Với những tiến bộ gần đây trong mạng nơ-ron sâu, Quá trình điểm thời gian sâu đã nổi lên như một cách tiếp cận đầy hứa hẹn để nắm bắt các mẫu phức tạp trong chuỗi sự kiện với dấu thời gian. Do đó, Quá trình điểm thời gian sâu có thể là một cách tiếp cận tiềm năng để dự đoán quá trình giám sát kinh doanh. Trong bài báo này, chúng tôi thử nghiệm và xem xét hiệu quả của các nghiên cứu gần đây trong Quá trình điểm thời gian sâu đối với vấn đề giám sát quy trình kinh doanh dự đoán. Kết quả của chúng tôi cho thấy rằng các phương pháp Quá trình điểm thời gian sâu có tiềm năng trong hoạt động tiếp theo và dự đoán thời gian còn lại trong dự đoán quy trình giám sát kinh doanh. Những phát hiện này có thể hữu ích cho các chuyên gia và nhà nghiên cứu quan tâm đến việc phát triển các hệ thống dự đoán giám sát cho các quy trình kinh doanh.

Từ khoá: *Giám sát quá trình kinh doanh, Quá trình điểm thời gian, Mạng nơ-ron sâu.*

Temporal point processes for business process monitoring

ABSTRACT

Predictive business process monitoring is a challenging time series task due to the complex and dynamic nature of business processes, which involves predicting the ongoing cases in terms of the next activity, activity suffix, and remaining time prediction on a business process. Temporal point processes (TPPs) are widely used to model sequences of events happening at irregular intervals, to model the occurrence times of events, and to capture the temporal dependencies among them. With the recent advances in deep neural networks, deep TPPs have emerged as a promising approach for capturing complex patterns in event sequences with occurrence timestamps. Hence, deep TPPs can be a potential approach to tackle business predictive monitoring tasks. In this paper, we experiment and review the effectiveness of recent research on deep TPPs on the predictive business process monitoring problem. Our results suggest that TPP methods have the potential in the next activity and remaining time prediction in the predictive business process monitoring problem. The findings can be helpful to practitioners and researchers interested in developing predictive monitoring systems for business processes.

Keywords: *Business Process Monitoring, Temporal Point Process, Deep Neural Network.*

1. INTRODUCTION

A business process is a collection of tasks performed asynchronously by various resources, such as humans, software, or hardware, to achieve a specific goal.[1] The execution of these tasks is tracked and documented in an event log, which records details such as the identifier of the case, the event performed, and the timestamp of the event.[2] There may also be optional case attributes, which are shared by events of the same case, or event attributes that are unique to each event. Business process mining is the discipline concerned with the analysis of these logs, tackling it from different perspectives such as discovering the underlying process model from the log, checking that the executions registered in the log are conformant with the process model, or extracting or inferring analytics that enhances the description of what has happened in the process executions.[3] Predictive monitoring is a process mining technique that predicts how an ongoing process case will unfold using the event log's information. The ability to make predictions is beneficial for anticipating issues before they arise, enabling the reallocation of resources before they are wasted, and providing recommendations.[4]

Studying the temporal distribution of events and discovering the relationships among different types of events is a great scientific approach for predictive monitoring and understanding the dynamics and mechanism of events occurrence.[5–9] One of its choices is the Temporal Point Process (TPP), the stochastic process with marked events on the continuous domain of time, which can naturally capture the clustering or self-correcting phenomena of such sequences of events.[10,11] Often, the rate of event occurrence, known as conditional intensity, is modeled as a function of time based on the prior observation of events to capture the dynamics of the process. Given that the conditional intensity function (CIF) entirely governs the distribution of such a process, statistical prediction and inference can all be performed via the CIFs.

Despite significant advancements in TPP, especially in models based on deep neural networks (DNNs), most of these models use different history encoders to embed historical events and various forms of intensity functions that are parameterized by the embedded historical sequence of events.[12–17] Also, to our knowledge, no experiment has been conducted on the efficiency of TPP in monitoring the business process.[18] Hence, in this paper, we compare different combinations of TPP methods regarding the history encoders and CIFs.

Hence, in this paper, our contribution is to define a data preprocessing procedure for the business process monitoring data set to suit the deep TPP models. We experiment to evaluate the capability and ability of deep TPP models on the predictive business process monitoring datasets.

2. BACKGROUND

This section provides an overview of the key concepts and techniques used in the study, including predictive business process monitoring and TPPs. It lays the foundation for understanding the experiments and results.

2.1. Predictive business process monitoring

The input of business process mining techniques is an event log, usually composed of events with at least a case identifier, an activity, and a timestamp, and, optionally, case attributes, which are values shared by all the events of the same case, and event attributes, which are specific of each event.[19,20] A sample log from the Helpdesk data set is shown in Table 1, part of a real-life help desk event log from an Italian Company.[21] This event log provides information about each event's case identifier, activity, timestamp, and resource.

Given a certain event prefix of a running case, predictive monitoring is concerned with forecasting how different aspects of the next event or sequence of events will unfold until the end of the case. There are several prediction targets, such as next activity, next activity suffix, next timestamp, next remaining time, next outcome, next attributes, and next attribute suffix.[22] In this paper, our interest is the next activity and the next timestamp prediction.

Formally, m_i, t_i, e_i is the activity, timestamp, and event. Let $hd^j(\sigma)$ be an event prefix such as $hd^j(\sigma) = \langle e_1, \ldots, e_j \rangle$. Two tackled problems can be defined as the following functions Ω using the newly predicted activities as new inputs for the next prediction until the dummy activity representing the end of the case *("[EOC]")* is reached:

- The next activity prediction problem: $\Omega_M\left(hd^j(\sigma)\right) = m'_{k+1}$.
- The next timestamp prediction problem: $\Omega_T\left(hd^j(\sigma)\right) = t'_{k+1}$.

2.2. Temporal point processes (TPPs)

2.2.1. Definition

Marked TPP is a random process representing as an event sequence $X = \{(t_1, m_1), \ldots, (t_N, m_N)\}$ with the increasing arrival times of events $\{t_i\}_{1\leq i\leq N}$ and markers $\{m_i\}_{1\leq i\leq N}$, such that $t_i \in [0, T), t_i < t_{i+1}, \forall i \ \geq 1$ where N is the number of events.

The mark is equivalent to the event's activity within the context of the business process. Thus, both terms can be used interchangeably afterwards. The inter-event time $\tau_i = t_i - t_{i-1}$ is also considered due to their convenience in computing.[15]

Categorical marks $\mathcal{M} = \{1,2, \ldots, K\}$ occurring in the time interval $[0, t)$ of the type-k event. The history $\mathcal{H}(t) = \{(t_j, m_j), t_j < t\}$ which can be considered the event prefix $hd^j(\sigma)$ in the business monitoring context.

The task of TPP models is to parameterize the K conditional intensity function (CIF) $\lambda_k^*(t)$, which can be characterized as follows:

$$\lambda_k^*(t) = \lambda_k\left(t\middle|\mathcal{H}(t)\right)$$
$$= \lim_{\Delta t \to 0^+} \frac{Pr(\text{ event of type } k \text{ in } [t, t + \Delta t) \mid \mathcal{H}_t\ )}{\Delta t}$$

which is defined as the expected instantaneous rate of happening events given the history. The $*$ symbol indicates the conditioning on the history $\mathcal{H}(t)$.

Due to the TPP modeling the distribution of the next timestamp t_i or inter-event τ_i time under the history $\mathcal{H}(t_i)$, the next timestamp prediction task is equivalent to considering the next timestamp t_i given $\mathcal{H}(t_i)$ denoted as follows:

$$\Omega_T\left(\mathcal{H}(t)\right) = P_i^*(t)$$

Given the CIF, the distribution $P_i^*(t)$ can be represented by any following functions:[12,15,16,23,24]

1. Probability density function (PDF): $f_i^*(t)$

Table 1. Excerpt of a Helpdesk's business process log

Case ID	Activity	Resource	Timestamp
Case 1	Assign seriousness	Value 1	2012/10/09 14:50:17
Case 1	Take in charge ticket	Value 1	2012/10/09 14:51:01
Case 1	Take in charge ticket	Value 2	2012/10/12 15:02:56
Case 1	Resolve ticket	Value 1	2012/10/25 11:54:26
Case 1	Closed	Value 3	2012/11/09 12:54:39
Case 2	Assign seriousness	Value 4	2012/04/03 08:55:38
Case 2	Take in charge ticket	Value 4	2012/04/03 08:55:53
Case 2	Resolve ticket	Value 4	2012/04/05 09:15:52
Case 2	Closed	Value 5	2012/05/19 09:00:28

2. Cumulative distribution function (CDF): $F_i^*(t) = \int_0^{t_i} f_i^*(u)du$
3. Survival function: $S_i^*(t) = 1 - F_i^*(t)$
4. Hazard function: $\phi_i^*(t) = f_i^*(t)/S_i^*(t)$
5. Cumulative hazard function (CHF): $\Phi_i^*(t) = \int_0^t \phi_i^*(u)du$

Here, we pick the PDF of the type-k event at time t as the parametric form, which is defined as:

$$f_k^*(t) = \lambda_k^*(t)\exp\left(-\int_{t_{i-1}}^{t} \lambda_k^*(u)du\right) \qquad (1)$$

where $\exp\left(-\int_{t_{i-1}}^{t} \lambda_k^*(u)du\right)$ is an exponential term where the exponent is the negative integral from t_{i-1} (the time of the last event before t, namely $i-1 = arg\,max_{j\leq n}\{t_j, t_j < t\}$) to t of the CIF. This integral represents the expected number of type-k events at time u, which happens in the time interval $(t_{i-1}, t]$. The exponential of the negative of this value penalizes the presence of other events in the interval, making it less likely for a new event to occur at time t. The product of these two parts gives the probability density of an event of type k occurring strictly at time t. Note that we must integrate this density over that interval to obtain the probability of an event occurring within a specific time interval. By aiming to parameterize a model to fit the timestamp distribution, the TPP can infer PDF or CIF for timestamp prediction, including the next event's timestamp and activity prediction.

2.2.2. Conditional Intensity Formulation (CIF)

The CIF with parameters $\Theta_k(t)$ is written as $\lambda_k\big(t; \Theta_k(t)\big|\mathcal{H}(t)\big)$. The parameter $\Theta_k(t)$ is considered a piece-wise function of t as:

$$\Theta_k(t) = \chi_k(h_i)$$

where $t \in [t_{i-1}, t_i)$. The formula means that the new occurrence of the type-k event changes the h_i and thus updates the $\Theta_k(t)$.

The choice of the family of CIF functions to approximate the target CIF is critical because the function's ability to approximate accurately determines the TPP's performance in fitting the distribution. Additionally, Equation (**1**) indicates that the integral term is unavoidable if we maximize the likelihood of the observed sequence of events. Hence, the challenge in computing the log-likelihood is the high computational cost due to the integral term. The closed form of this integral term, such as cumulative hazard function, can make the computation of likelihood feasible.[15,23,24] In conclusion, the goal of approximating the target CIF is to choose a family of functions in the closed integral form with powerful expressivity.

2.2.3. Modeling the marks

In the business monitoring context, where multiple event types exist, the next activity prediction task is determining the most probable event type based on historical data, which can be dealt with as the categorical classification task. It is generally achieved by first converting the historical encoding to logit scores of a discrete distribution, as shown in the following equation:

$$\kappa(h_i) = \text{logit}(\widehat{m_i}) \qquad (2)$$

where $\text{logit}(\widehat{m_i}) \in R^K$, $\kappa: R^D \to R^K$.

Then, we apply a softmax function to transform logit scores into the categorical distribution, which is the solution to the next activity prediction task as follows:

$$\Omega_M\big(\mathcal{H}(t)\big) = \Pr\big(\widehat{m_i} = k\big|\mathcal{H}(t)\big)$$
$$= \text{softmax}\big(\text{logit}(\widehat{m_i})\big)_k$$

where $\text{softmax}\big(\text{logit}(\widehat{m_i})\big)_k$ is to choose the k-th mark from its output. By forming the loss of activities as the logit scores, the cross-entropy loss for categorical classification is added to the log-likelihood loss given the actual activity m of the i-th event to maximize the joint likelihood of the next timestamp and activity, which is considered independent. Several works on maximizing joint likelihood in conditional forms are proposed, such as time conditioned on marks[25,26] and marks conditioned on time[15], which can capture the dependencies between timestamp and activity and leverage the TPP models performance in predicting the timestamp and activity simultaneously. In our experiment, we utilize the idea of using the joint negative log-likelihood (NLL) under the independence between the next timestamp and activity, with the type-k mark for a single sequence X for categorical marks computed as:

$$\begin{aligned} &-\log p\,(X) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} -\Pr\big(\widehat{m_i} = k\big|\mathcal{H}(t)\big) \qquad (3) \\ &+ \sum_{k=1}^{K} f_k^*(t) \end{aligned}$$

3. CLASSIFICATION OF TPP MODELS

This section introduces the classification of TPP models to solve the predictive business process monitoring problem. As illustrated in Figure 1, our procedure considers two essential parts of a deep TPP: the history encoder and the CIF.

Table 2. The classification of all options for each component by history encoder, mixture distribution, and prediction target.

History Encoder	Mixture Distribution	Prediction
Recurrent neural network (RNN)	Log-Normal	Next activity
Gated recurrent unit (GRU)	Gompertz	Next timestamp
Long short-term memory (LSTM)	Log-Cauchy	
Attention	Exponential decay	
Fourier Transformer (FNet)	Weibull	

3.1. Historical Event Encoders

Since CIF or PDF is a function of t and historical events before t, namely $\mathcal{H}(t)$, we have to encode the history sequence of each event (t_j, m_j) as a feature vector e_j to formulate the CIF or PDF of the occurrence of different events to model the process. For the i-th event's history, $\mathcal{H}(t_i)$, j-th event in the history set is embedded in a high-dimensional space including time and mark features, as follows:

$$e_j = [\omega(t_j); E^{T m_j}]$$

where:

- ω represents the time feature that transforms one-dimension temporal information t_j (or inter-event time τ_j) into a high-dimension vector directly or via its logarithm[16,26] or trigonometric functions[12,27].
- E represents the mark feature, an embedding matrix for marks, and m_j is the one-hot encoding of mark m_j.

A historical encoder H can be obtained via concatenation of the sequence of embedding $\{e_1, e_2, \ldots, e_{i-1}\}$ into a vector space of dimension D under the following formula:

$$h_i = H(\{e_1; e_2; \ldots; e_{i-1}\})$$

H can be chosen as Recurrent-based encoders, Attention-based encoders, or Fourier transform encoders, and h_i is utilized for the CIF parameterization.

3.1.1. Recurrent-based encoders

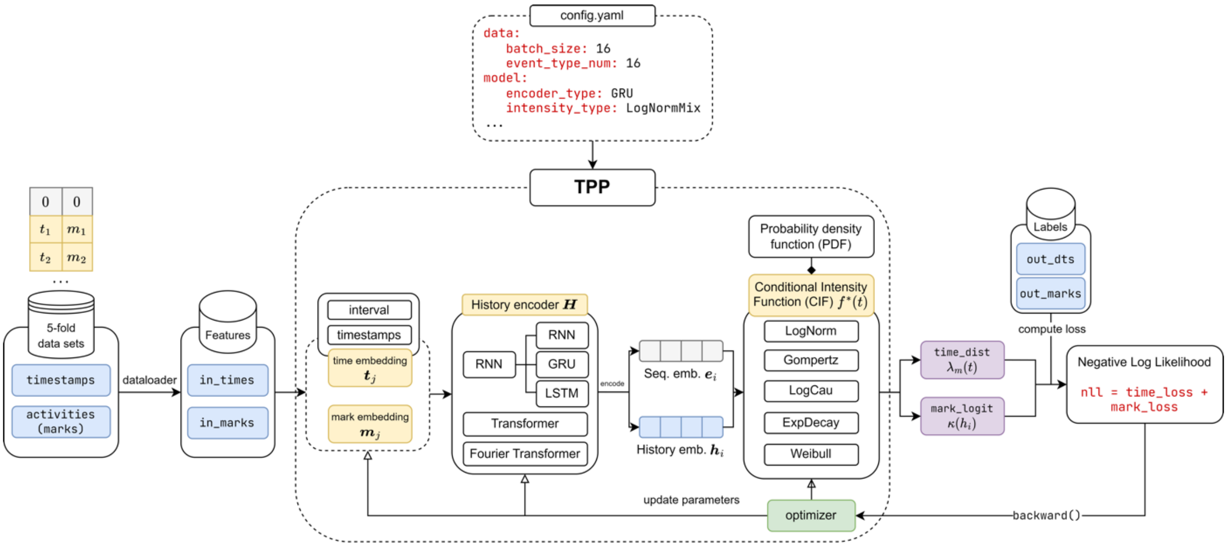
Recurrent-based encoders, including RNN units, GRU, and LSTM, can be used as history encoders.[14–16] Their CIF can be formulated as follows:

$$h_0 = 0; \quad h_i = \mathrm{RNN}(e_{i-1}, h_{i-1})$$

where the initial state of the history encoder, h_0, is set as zero. For each subsequent time step i, the new state, h_i, is updated based on the previous state, h_{i-1}, and the previous event, e_{i-1}, through the RNN function. The RNN takes as inputs the previous state and the previous event and outputs the new state. This state represents the RNN's memory, encoding information about past events that it can use to predict future events.

The advantage of using recurrent-based encoders as the history encoder is that it requires low storage space due to the capability of serial computing. The states and events are processed one at a time, meaning that the RNN does not need to store all of them at once, which can be beneficial in situations where storage space is limited.

However, there are also disadvantages to this approach. The serial computing nature of RNNs can limit their computational speed in both the forward and backward processes.[28] Additionally, RNNs can suffer from issues such as the gradient vanishing effect, where the gradients used in learning become very small, making learning slow or even impossible.[29] They can also suffer from long-term memory loss, struggling to retain information about events that occurred long


config.yaml

```
data:
   batch_size: 16
   event_type_num: 16
model:
   encoder_type: GRU
   intensity_type: LogNormMix
...
```

Figure 1. The procedure of TPPs framework for predictive business process monitoring.

ago. These issues can potentially compromise the performance of the RNN history encoder.

3.1.2. Attention-based encoders

Attention-based encoders are part of encoder-decoder architectures that utilize the concept of attention. This mechanism allows models to focus on relevant parts of the input sequence when generating an output.[30] Self-attention is proposed as the history encoder in TPPs with fast parallel computing and the capability of encoding more long-term sequences than recurrent-based encoders.[12] The attention-based history encoders for CIF can be defined as follows:

$$h_i = \frac{\sum_{j=1}^{i-1} \phi(e_j, e_{i-1})\psi(e_j)}{\sum_{j=1}^{i-1} \phi(e_j, e_{i-1})}$$

where h_i represents the encoded history at time i. This history is computed as a weighted sum of transformed event embeddings $\psi(e_j)$, where the attention mechanism determines the weights $\phi(e_j, e_{i-1})$. The attention mechanism $\phi(\cdot,\cdot)$ is a function that takes two event embeddings as inputs and outputs a scalar called the attention weight. This weight determines the importance or relevance of the event e_j when encoding the history at the time i. The transformation function ψ transforms the event embedding e_j into a series of D-dimensional vectors called values. These transformed embeddings are then used to compute the encoded history.

While attention-based encoders overcome some of the problems with RNNs, their space complexity of the attention matrix is $O(N^2)$, which can become problematic when dealing with very long sequences because the attention mechanism computes pairwise interactions between all events, leading to a quadratic increase in storage requirements as the number of events increases.[31] This problem can be temporarily resolved by limiting the encoder only to access the last L events $\{y_{i-L}, \dots, y_{i-1}\}$, which can reduce the time complexity to $O(NL)$.[15]

3.1.3. Fourier transform encoders

The Fast Fourier Transform (FFT) module was generally used in the natural language processing (NLP) field[32] and recently adopted into the history encoder family under the TPP context called FNet, which aims to speed up the computation and replace the attention mechanism.

$$h_i = \text{Top}_p\{\text{FFT}([\text{FFT}(e_1); \dots; \text{FFT}(e_{i-1})])\}$$

where the $\text{FFT}(\cdot)$ represents the FFT, which operates on the events' embedding, then on the whole sequence. Then, $\text{Top}_p\{\cdot\}$ means choosing the highest p frequencies in the set as the history encoding. The dimension of the feature vector e_j has to equal D dimension. Top-p needs to be chosen due to the unequal event embedding history sequence length, so the padding operation is required for batch processing as many sequences contain the same padding values, which contain useless information and lead to low-frequency values in spectra. Therefore, filtering the low frequency can capture and retain more information about the historical sequence in the high frequency.

FFT encoder inherits fast computational time complexity in $O(NlogN)$ and the ability to capture long-term patterns due to the global property of the sequences' spectrum.[33] However, one disadvantage of this approach is that the backward process of gradient propagation leads to significant memory complexity.

3.2. Mixture Distribution

The mixture distribution family is the major component in TPP that approximates the target PDF under CIF.

3.2.1. Log-normal Mixture

Log-normal Mixutre is proposed to approximate any distribution due to the feasible computation of its PDF and cumulative distribution function (CDF) in the closed form of CIF and CHF, where the mixture form reads as follows:[15]

$$\begin{aligned} &f^*_{\mathrm{LNM}}(t) \\ &= \sum_{s=1}^{S} w_s \frac{1}{t\sigma_{s\sqrt{2\pi}}} \exp\left(-\frac{(\ln(t-t_{i-1})-\mu_s)^2}{2\sigma_s^2}\right) \end{aligned}$$

where $t \in [t_{i-1}, t_i)$. S are mixture distribution numbers, $\{w_s\}_{1\leq s\leq S}$ are non-negative mixture weights and $\sum_{s=1}^{S} w_s = 1$, $\sigma_s > 0 \forall s$.

The distribution of the next timestamp $\lambda^*_{\mathrm{LN}}(t)$ under the log-normal mixture can be modeled by different functions such as PDF, CDF, survival function, hazard function, or cumulative hazard function. Moreover, the preferable function to model the distribution $\lambda^*_{\mathrm{LN}}(t)$ is the cumulative hazard function due to its ability to compute the NLL in the closed form without numerical integration, leading to the loss function in Eq. (**3**) replaces the $f^*_m(t)$ by the $\Phi^*_m(t)$. Although the CDF has no closed form, the approximation of the function has minor deviation and permits gradient back-propagation, allowing both the forward and backward processes.

3.2.2. Gompertz Mixture

Gompertz Mixture is proposed to predict both timestamps and marks of future events without any prior knowledge about the hidden functional forms of the latent temporal dynamics.[14] The CIF of Gompertz distribution reads as:

$$\lambda(t) = \eta \exp(\beta t)$$

where $\eta, \beta > 0$. The corresponding PDF can be obtained as follows:

$$\lambda^*_{\mathrm{GP}}(t) = \exp(\beta(t-t_{i-1}) + v^T h_i + b_t)$$

which its PDF reads:

$$\begin{aligned} f^*_{\mathrm{GP}}(t) = \eta \exp\Bigg(&\beta(t-t_{i-1}) \\ &-\frac{\eta}{\beta}\big(\exp\big(\beta(t-t_{i-1})\big)-1\big)\Bigg) \end{aligned}$$

for $t \in [t_{i-1}, t_i)$, where $\eta = \exp\big(v^{Th_i} + b_t\big)$, and $\beta > 0$. The process $f^*_{\mathrm{GP}}(t)$ becomes the Poisson distribution when $\beta = 0$. The mixture can be formulated as follows:

$$\begin{aligned} f^*_{\mathrm{GPM}}(t) = \sum_{s=1}^{S} w_s \, \eta_s \exp\Bigg(&\beta_s(t-t_{i-1}) \\ &-\frac{\eta_s}{\beta_s}\big(\exp\big(\beta_s(t-t_{i-1})\big)-1\big)\Bigg) \end{aligned}$$

for $t \in [t_{i-1}, t_i)$, where $\beta_s > 0$ and $\eta_s > 0$ for any s. The parameters are obtained as a function of history encoding h_i for $t \in [t_{i-1}, t_i)$ as follows:

$$\Theta(t) = \{w_s(t), \beta_s(t), \eta_s(t)\}_{1\leq s\leq S} = \chi(h_i)$$

3.2.3. Exp-decay Mixture

Zhang et al.[12] extend the expressivity of the multivariate Hawkes Process by the Self-Attentive Hawkes Process (SAHP) by adapting the self-attention mechanism to fit the intensity function of the Hawkes processes. This allows the Exp-decay mixture to capture longer historical information and is more interpretable because the learned attention weight tensor shows the contributions of each historical event. It models the intensity function as the exponential-decaying form like the classical Exp-decay Hawkes Process and extends with a nonlinear transform $softplus$ stacked after. This is also the cause of unmanageable computation of the integral term when dealing with long data sequences due to attention weights computation for each pair of events in the sequence under the self-attention mechanism.

To solve the infeasible computation, the final transformation of non-linearity is removed, and the CIF of the defined Exp-decay distribution can be obtained as follows:

$$\lambda^*_{\mathrm{ED}}(t) = \eta \exp\big(-\beta(t-t_{i-1})\big) + \alpha$$

where the first term indicates the impacts of historical events decay with an exponential ratio added with the α, which is the basic intensity.

By using the distribution as a component, the mixture of Exp-decay distribution reads the PDF as follows:

$$\begin{aligned} f^*_{\mathrm{EDM}}(t) = \sum_{s=1}^{S} w_s\big(&\eta_s \exp\big(-\beta_s(t-t_{i-1})\big) \\ &+\alpha_s\big) \exp\Bigg(\bigg(\frac{\eta_s}{\beta_s} \\ &-1\bigg) \exp\big(-\beta_s(t-t_{i-1})\big) \\ &-\alpha_s(t-t_{i-1})\Bigg) \end{aligned}$$

for $t \in [t_{i-1}, t_i)$, whose parameters are all positive, calculated by $\chi(h_i)$.

3.2.4. *Weibull Mixture*

Weibull Mixture assumes a population of two or more subpopulations with different Weibull distributions.[34] The Weibull distribution has the advantage of high approximating ability due to no numerical instability, so the parameter range is not limited to a certain range. The parameters of the Weibull mixture model can be estimated using the maximum likelihood estimation (MLE) or Bayesian methods. Its CIF reads:

$$\lambda^*_{\mathrm{WB}}(t) = \eta\beta\big(\eta(t - t_{i-1})\big)^{\beta-1}$$

where $\eta, \beta > 0$. CIF will increase when $\beta > 1$, decrease when $\beta < 1$, and be constant when $\beta = 1$.

And its PDF represents:

$$f^*_{\mathrm{WBM}}(t) = \sum_{k=1}^{S} w_s \eta_s \beta_s \big(\eta_s(t - t_{i-1})\big)^{\beta-1} \exp\left(-(\eta_s(t - t_{i-1})^{\beta})\right)$$

3.2.5. *Log-Cauchy Mixture*

Log-Cauchy Mixture is utilized because the Log-Cuachy distribution can model a wide range of data due to its flexibility by handling both symmetric and asymmetric data and super-heavy-tailed distributions with no given mean or standard deviation.[35] The Log-Cauchy distribution is also robust to outliers, which is helpful in monitoring the business.[36] The Log-Cauchy mixture with the PDF is written as:

$$f^*_{\mathrm{LCM}}(t) = \sum_{s=1}^{S} \frac{w_s}{(t - t_i)\pi} \frac{\sigma}{(\ln(t - t_i) - \mu)^2 + \sigma^2}$$

4. EXPERIMENTS

This section details the experimental setup, including the datasets used, the procedure followed, and the evaluation metrics employed for next activity prediction and next timestamp prediction tasks. This section also provides the necessary context for interpreting the results.

4.1. Dataset

We evaluate two real-time event logs extracted from the *4TU Center for Research Data* to evaluate different combinations of history encoders and conditional intensities: Helpdesk[21] and BPI 2012[37]. Table 3 shows relevant statistics from these logs, namely, the number of cases, the number of different activities, the number of events, the average and maximum case length, the maximum and mean event duration in days, the mean and maximum case duration in days and the number of different variants.

Table 3. Statistics of the event logs used for benchmarking. Time-related measures are shown in days.

Statistics	**Helpdesk**[21]	**BPI 2012**[37]
Number of cases	4580	13087
Number of event types	14	36
Number of events	21348	262200
Mean case length	4.66	20.04
Maximum case length	15	175
Mean event duration	11.16	0.45
Maximum event duration	59.92	102.85
Mean case duration	40.86	8.62
Maximum case duration	59.99	137.22
Variants	266	4366

4.2. Procedure

Figure 1 shows the procedure of our framework from preprocessing the data sets, data splitting, and training the history encoder and CIF until inferring the timestamps and activities. We perform data splitting and evaluate all TPP combinations in identical conditions to ensure comparable results. The goal is to simulate a scenario where past knowledge is utilized for training a predictive model, which is then used to predict the future. To achieve this, we perform a 5-fold cross-validation, where every approach is tested once per fold. The event log traces are sorted by their initial event timestamp and split into training, validation, and test sets with a distribution of 64%, 16%, and 20%, respectively. Timestamps and activities are extracted from each fold data set and then encoded into history embeddings h_i and sequence embeddings e_i by a chosen history encoder H. These embeddings are modeled under a specific CIF $f^*(t)$ to extract the time distribution and mark logit as the next timestamp and activity predictions. Later, we utilize the ground truth to compute the joint NLL (Eq. (**3**)) and optimize the parameters for the embeddings, the encoder, and the CIF.

4.3. Metrics

We use the following metrics to evaluate the TPP combinations' performance on the next activity prediction and next timestamp prediction tasks, along with the goodness-of-fit by NLL as Eq. (**3**).

The results are reported as the mean performance of each TPP combination on five folds. Additionally, our evaluation appends a dummy event as an *(“[EOC]”)* token to the end of every log trace, which can reduce the process state and provide a clear stopping point for activity prediction.

4.3.1. Next activity prediction

We use the accuracy metric since the next activity prediction task is a classic classification problem. The accuracy measures the proportion of correct classifications in relation to the number of predictions done, which is implemented as follows:

$$\text{Top-}q\ \text{ACC}(\{\widehat{m_i}\}_{1\le i\le N},\{m_i\}_{1\le i\le N}) = \frac{\left|\{m_i \in \text{Top}_q\{\text{logit}(\widehat{m_i})\}: 1 \le i \le N\}\right|}{N}$$

where $\text{logit}(\widehat{m_i}) \in R^K$ is obtained by Eq. (**2**) to measure the predicted discrete probability.

4.3.2. Next timestamp prediction

Since the time prediction problem is a regression task, the metric chosen for measuring the TPP performance in the next timestamp prediction task is the Mean Absolute Error (MAE). Instead of evaluating TPP performance based on the normalized value taken directly from the distribution under the Mean Absolute Percentage Error (MAPE), we alter the normalization step and postprocessing step to return the next timestamp prediction in days to have a fair comparison with our benchmarks. MAE metric has the advantage of not over-penalizing the variability in the observations, which is important in the time prediction in predictive process monitoring, where the time between two events in a trace can be potentially large. The MAE is defined as follows:

$$\text{MAE}(\{\widehat{t_i}\}_{1\le i\le N},\{t_i\}_{1\le i\le N}) = \frac{\sum_{i=1}^{N}|\widehat{t_i} - t_i|}{N}$$

where $\widehat{t_i}$ is the i-th predicted timestamp.

Table 4. Experimental results of the next timestamp prediction MAE in days and the next activity accuracy of modeling the overall CIF with different combinations of history encoder and family of distribution. The arrows ↑/↓ indicate that the higher/lower results, the better. The metrics are computed as the mean of the 5-fold cross-validation. The metrics in **bold** mean that the model achieves the top-5 performance in the column.

Methods	Helpdesk				BPI 2012			
	NLL ↓	**MAE ↓**	**Top-1 ACC ↑**	**Top-3 ACC ↑**	**NLL ↓**	**MAE ↓**	**Top-1 ACC ↑**	**Top-3 ACC ↑**
LogNormMix+RNN	**-2.046201**	279.991608	0.695686	0.814908	**-4.419353**	19.107306	0.809411	0.934645
LogNormMix+GRU	**-2.049259**	269.787903	0.696841	0.814715	**-4.498922**	19.963001	0.811243	**0.935909**
LogNormMix+LSTM	-1.965748	265.744049	**0.698613**	0.814715	**-4.805172**	17.676338	0.810400	0.935561
LogNormMix+Attention	**-2.025522**	243.183641	**0.698927**	0.814715	**-4.970017**	19.699347	0.804081	0.942155
LogNormMix+FNet	**-2.088489**	354.522430	0.697804	0.814715	-3.997465	17.554281	0.700791	0.924937
GomptMix+RNN	0.862758	26.262592	0.694530	**0.815293**	-1.943585	3.082433	0.811463	0.935542
GomptMix+GRU	0.756814	36.176449	**0.698190**	0.814522	-1.989259	3.715980	**0.814430**	**0.935689**
GomptMix+LSTM	0.803448	33.733837	0.694915	**0.814909**	-1.718078	**1.786893**	**0.811939**	0.935103
GomptMix+Attention	0.198063	**24.308924**	**0.798151**	**0.974923**	-2.651078	**2.053919**	0.810492	**0.936513**
GomptMix+FNet	0.922608	26.557230	0.695300	0.812982	-1.338134	**1.192549**	0.658058	0.919295
LogCauMix+RNN	-0.245449	**24.777443**	0.697612	**0.815100**	-2.425044	**1.674708**	0.809997	0.933234
LogCauMix+GRU	-0.238866	25.448597	0.693374	0.812789	-2.447077	4.090276	0.809778	0.935616
LogCauMix+LSTM	-0.213667	**9.497505**	0.697034	0.811633	-2.442514	2.642548	0.810345	0.935744
LogCauMix+Attention	-0.572755	**20.529713**	0.697612	0.812982	-2.426620	**1.027533**	0.806151	0.928967
LogCauMix+FNet	-0.229975	**24.389563**	**0.698960**	0.814137	-2.372896	2.651052	0.698978	0.924735
WeibMix+RNN	-1.920710	354.471039	0.690100	0.814522	-4.106110	17.718061	0.807818	0.933308
WeibMix+GRU	-1.807741	35.020641	0.689137	0.811633	**-4.190582**	14.316733	0.808514	0.933930
WeibMix+LSTM	-1.856559	295.800934	0.686248	0.811633	-4.111487	15.760619	**0.812818**	0.935542
WeibMix+Attention	**-1.986018**	123.920174	0.687982	0.812982	-4.050585	19.363649	0.796498	0.931842
WeibMix+FNet	-1.750751	308.825226	0.683166	0.808166	-3.867040	19.018917	0.694252	0.923929
ExpDecayMix+RNN	1.778744	67.937225	0.696263	0.814908	1.221636	20.735715	0.811005	0.935011
ExpDecayMix+GRU	1.889372	117.146721	0.697612	**0.815293**	-1.230995	20.580959	**0.814375**	**0.935689**
ExpDecayMix+LSTM	1.963200	85.079498	0.696456	0.813367	-1.205069	20.735718	**0.812086**	0.935231
ExpDecayMix+Attention	1.447936	354.209351	0.696841	0.814522	-1.521961	20.748582	0.808111	0.934700
ExpDecayMix+FNet	1.800676	351.875275	0.697997	0.814522	-1.036910	20.735718	0.699857	0.924918

5. RESULTS

This section presents the findings from the experiments, highlighting the performance of different TPP models based on their combinations of history sequence encoders and formulations of conditional intensity functions. This section summarizes the key observations and insights gained from the experiments.

Table 4 evaluates different combinations of history encoders and overall conditional intensities on two real-world datasets, namely Helpdesk and BPI 2012.

- **Goodness-of-fit** is typically evaluated via the NLL result. The choice of history encoders such as RNN-based, Attention-based, and FNet-based methods usually does not affect the overall performance of TPP models regarding the goodness-of-fit. Meanwhile, the intensity functions used for CIF approximation matter most. ExpDecayMix shows the worst fitting ability. Besides, LogNormMix and WeibMix usually fit the data best due to the ability to fit the distribution via the NLL.
- **Next timestamp prediction** is evaluated according to the MAE. The choice of intensity function is also crucial, where LogCauMix and GomptMix usually predict significantly better than others. In the Helpdesk dataset, interestingly, the LSTM with LogCauMix performs far better than any other combinations of TPP models.
- **Next activity prediction** is evaluated via the Top-1 ACC and Top-3 ACC. The results show that the history encoder is critical because the prediction depends on its encodings. Attention-based encoders usually have good predictive performance because they can capture long-term features from historical events. Besides, GRU and LSTM also achieve high results due to their ability to capture long memory.

To sum up, the NLL and MAE calculated by timestamps are predominantly influenced by the formulation of intensity and short-term influences, which the five history encoders can adequately capture. Though FNet is a new proposed approach and does not achieve high results compared with other history encoders, it still shows potential when pairing with suitable intensity functions such as LogCauMix. All history encoders can sufficiently model the dynamics of arrival time, given their minor differences. In contrast, Attention-based encoders usually surpass other history encoders to model the dynamics of the next activity due to the capability of capturing the long-term impacts of historical events.

6. DISCUSSION

The experimental results presented in this study provide valuable insights into the performance of different TPP models for predictive business process monitoring tasks. The findings highlight the importance of selecting an appropriate combination of history sequence encoders and CIFs to achieve optimal results. One key observation is that the choice of CIF plays a crucial role in the next timestamp prediction task. The LogCauMix and GomptMix intensity functions consistently outperform other options, indicating their suitability for capturing the temporal dynamics of business processes. This suggests that the formulation of the intensity function should be carefully considered when designing TPP models for timestamp prediction. Another notable finding is the impact of the history encoder on the next activity prediction task. Attention-based encoders, such as the self-attention mechanism, demonstrate superior performance compared to other encoders. This can be attributed to their ability to capture long-term dependencies and selectively focus on relevant historical events. The results underscore the importance of leveraging attention mechanisms to effectively model the complex relationships between past activities and future predictions. The experiments also reveal that the FNet encoder, despite being a relatively new approach, shows potential when paired with suitable intensity functions like LogCauMix. While its performance may not surpass other established encoders, the FNet's ability to capture temporal patterns efficiently makes it a promising direction for future research in TPP models for business process monitoring. Conclusively, the performance of TPP models can vary depending on the characteristics of the dataset and the specific business process being monitored. The Helpdesk and BPI 2012 datasets used in this study represent real-world scenarios, but the generalizability of the findings to other domains and processes should be further investigated. Future research could explore the application of TPP models to a wider range of business processes and datasets to validate the robustness of the observed trends.

Another important aspect to consider when applying TPP models in predictive business process monitoring is explainability.[38] As businesses increasingly rely on automated decision-making systems, the ability to interpret and understand the predictions made by these models becomes crucial. Explainability helps to build trust in the model's outputs, facilitates debugging and error analysis, and enables stakeholders to gain insights into the factors influencing the predictions.[39–41] However, it is important to note that achieving explainability in TPP models is not without challenges. The complexity of the models, the high-dimensional nature of the input data, and the temporal dependencies can make it difficult to provide simple and intuitive explanations. Striking a balance between model performance and interpretability is an ongoing research challenge. Future work in this area could focus on developing novel explainability techniques tailored to TPP models, as well as conducting user studies to assess the effectiveness and usability of these explanations in real-world business settings.

7. CONCLUSION

In this paper, we evaluate the performance of different TPP models via their combinations of the history sequence encoder and formulation of CIF on the predictive business process monitoring data sets. The results show that the formulation of intensity influences the next timestamp prediction and can be captured by any of the history encoders. The next event prediction is dominated by the ability to capture long-term impacts from historical events, especially attention-based encoders. In our future work, we plan to conduct a more profound experiment around several aspects of TPP models, such as loss computation, history embedding's normalization, the relational discovery of events, and optimizations. In our future work, we extend the capability of TPP models on other prediction problems such as activity suffix and remaining time prediction, and continue research on explainability to bridge the gap between model performance and interpretability, ultimately leading to more effective and user-friendly monitoring systems.

REFERENCES

1. M. Kirchmer, others. *High performance through business process management*, Springer, 2017.
2. W. Van Der Aalst. *Process mining: discovery, conformance and enhancement of business processes*, Springer, 2011, (*2*).
3. W. Van Der Aalst. *Process mining: Overview and opportunities*, , *ACM Transactions on Management Information Systems (TMIS)*, **2012**, (*3*), 1–17.
4. F. M. Maggi, C. Di Francescomarino, M. Dumas, C. Ghidini. Predictive monitoring of business processes, *Advanced Information Systems Engineering: 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014. Proceedings 26*, **2014**, 457–472.
5. D. Luengo, M. Sepúlveda. Applying clustering in process mining to find different versions of a business process that changes over time, *Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I 9*, **2012**, 153–158.
6. H. R'bigui, C. Cho. *The state-of-the-art of business process mining challenges*, , *International Journal of Business Process Integration and Management*, **2017**, (*8*), 285–303.
7. Y. Bertrand, S. Veneruso, F. Leotta, M. Mecella, E. Serral. NICE: the Native IoT-centric event log model for process mining, *International Conference on Process Mining*, **2023**, 32–44.
8. M. Vidgof, S. Bachhofner, J. Mendling. Large language models for business process management: Opportunities and challenges, *International Conference on Business Process Management*, **2023**, 107–123.
9. A. Saha, P. Agarwal, S. Ghosh, N. Gantayat, R. Sindhgatta. Towards Business Process Observability, *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, **2024**, 257–265.
10. A. G. Hawkes. *Spectra of Some Self-Exciting and Mutually Exciting Point Processes*, , *Biometrika*, **1971**, (*58*), 83–90.
11. V. Isham, M. Westcott. *A self-correcting point process*, , *Stochastic Processes and their Applications*, **1979**, (*8*), 335–347.
12. Q. Zhang, A. Lipani, O. Kirnap, E. Yilmaz. Self-Attentive Hawkes Process, *Proceedings of the 37th International Conference on Machine Learning*, **2020**, (*119*), 11183–11193.

13. J. Yan, X. Liu, L. Shi, C. Li, H. Zha. Improving Maximum Likelihood Estimation of Temporal Point Process via Discriminative and Adversarial Learning, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, **2018**, 2948–2954.

14. N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, L. Song. Recurrent marked temporal point processes: Embedding event history to vector, *In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, **2016**.

15. O. Shchur, M. Biloš, S. Günnemann. *Intensity-Free Learning of Temporal Point Processes*, , *International Conference on Learning Representations (ICLR)*, **2020**.

16. T. Omi, naonori ueda, K. Aihara. Fully Neural Network based Model for General Temporal Point Processes, *Advances in Neural Information Processing Systems*, **2019**, (*32*).

17. H. Mei, J. M. Eisner. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process, *Advances in Neural Information Processing Systems*, **2017**, (*30*).

18. P. Gamallo-Fernandez, E. Rama-Maneiro, J. C. Vidal, M. Lama. *VERONA: A python library for benchmarking deep learning in business process monitoring*, , *SoftwareX*, **2024**, (*26*), 101734.

19. W. M. Van Der Aalst, H. A. Reijers, A. J. Weijters, B. F. van Dongen, A. A. De Medeiros, M. Song, H. Verbeek. *Business process mining: An industrial application*, , *Information systems*, **2007**, (*32*), 713–732.

20. D. Dakic, D. Stefanovic, I. Cosic, T. Lolic, M. Medojevic. *BUSINESS PROCESS MINING APPLICATION: A LITERATURE REVIEW.*, , *Annals of DAAAM & Proceedings*, **2018**, (*29*).

21. M. Polato. Dataset belonging to the help desk log of an Italian Company, **2017**.

22. E. Rama-Maneiro, J. Vidal, M. Lama. *Deep learning for predictive business process monitoring: Review and benchmark*, , *IEEE Transactions on Services Computing*, **2021**.

23. M. Okawa, T. Iwata, T. Kurashima, Y. Tanaka, H. Toda, N. Ueda. Deep mixture point processes: Spatio-temporal event prediction with rich contextual information, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, **2019**, 373–383.

24. A. Soen, A. Mathews, D. Grixti-Cheng, L. Xie. UNIPoint: Universally Approximating Point Processes Intensities, *Proceedings of the AAAI Conference on Artificial Intelligence*, **2021**, (*35*), 9685–9694.

25. J. Enguehard, D. Busbridge, A. Bozson, C. Woodcock, N. Hammerla. Neural temporal point processes for modelling electronic health records, *Machine Learning for Health*, **2020**, 85–113.

26. S. Zuo, H. Jiang, Z. Li, T. Zhao, H. Zha. Transformer hawkes process, *International conference on machine learning*, **2020**, 11692–11702.

27. S. Zhu, S. Li, Z. Peng, Y. Xie. *Imitation learning of neural spatio-temporal point processes*, , *IEEE Transactions on Knowledge and Data Engineering*, **2021**, (*34*), 5391–5402.

28. Y. Su, C.-C. J. Kuo, others. *Recurrent neural networks and their memory behavior: a survey*, , *APSIPA Transactions on Signal and Information Processing*, **2022**, (*11*).

29. P. Le, W. Zuidema. Quantifying the Vanishing Gradient and Long Distance Dependency Problem in Recursive Neural Networks and Recursive LSTMs, *Proceedings of the 1st Workshop on Representation Learning for NLP*, **2016**, 87–93.

30. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. *Attention is all you need*, , *Advances in neural information processing systems*, **2017**, (*30*).

31. K. M. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Q. Davis, A. Mohiuddin, L. Kaiser, others. Rethinking Attention with Performers, *International Conference on Learning Representations*, **2020**.

32. J. Lee-Thorp, J. Ainslie, I. Eckstein, S. Ontanon. FNet: Mixing Tokens with Fourier Transforms, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, **2022**, 4296–4313.

33. H. Lange, S. L. Brunton, J. N. Kutz. *From fourier to koopman: Spectral methods for long-term time series prediction, , The Journal of Machine Learning Research*, **2021**, (*22*), 1881–1918.

34. J. M. Marín, M. T. Rodríguez-Bernal, M. P. Wiper. *Using Weibull Mixture Distributions to Model Heterogeneous Survival Data, , Communications in Statistics - Simulation and Computation*, **2005**, (*34*), 673–684.

35. T. Ruzgas, M. Lukauskas, G. Čepkauskas. *Nonparametric multivariate density estimation: case study of cauchy mixture model, , Mathematics*, **2021**, (*9*), 2717.

36. Z. I. Kalantan, J. Einbeck. *Quantile-based estimation of the finite cauchy mixture model, , Symmetry*, **2019**, (*11*), 1186.

37. B. van Dongen. BPI Challenge 2012, **2012**.

38. J. N. Adams, S. J. van Zelst, T. Rose, W. M. van der Aalst. *Explainable concept drift in process mining, , Information Systems*, **2023**, (*114*), 102177.

39. N. Mehdiyev, P. Fettke. *Explainable artificial intelligence for process mining: A general overview and application of a novel local explanation approach for predictive process monitoring, , Interpretable artificial intelligence: A perspective of granular computing*, **2021**, 1–28.

40. H. T. T. Nguyen, H. Q. Cao, K. V. T. Nguyen, N. D. K. Pham. Evaluation of explainable artificial intelligence: Shap, lime, and cam, *Proceedings of the FPT AI Conference*, **2021**, 1–6.

41. T. T. H. Nguyen, V. B. Truong, V. T. K. Nguyen, Q. H. Cao, Q. K. Nguyen. Towards trust of explainable ai in thyroid nodule diagnosis, *International Workshop on Health Intelligence*, **2023**, 11–26.