Computers and Education: Artificial Intelligence 4 (2023) 100133

Contents lists available at ScienceDirect

Computers and Education: Artificial Intelligence

journal homepage: www.sciencedirect.com/journal/computers-and-education-artificial-intelligence

Prediction of user temporal interactions with online course platforms using deep learning algorithms

Junru Ren, Shaomin Wu *

Kent Business School, University of Kent, Canterbury, Kent CT2 7FS, UK

ARTICLE INFO

Keywords:
User action analysis
Temporal interaction prediction
Deep learning in education
Marked temporal point process
Recurrent neural networks

ABSTRACT

The analysis of learning interactions during online studying is a necessary task for designing online courses and sequencing key interactions, which enables online learning platforms to provide users with more efficient and personalized service. However, the research on predicting the interaction itself is not sufficient and the temporal information of interaction sequences hasn't been fully investigated. To fill in this gap, based on the interaction data collected from Massive Open Online Courses (MOOCs), this paper aims to simultaneously predict a user's next interaction and the occurrence time to that interaction. Three different neural network models: the long short-term memory, the recurrent marked temporal point process, and the event recurrent point process, are applied on the MOOC interaction dataset. It concludes that taking the correlation between the user action and its occurrence time into consideration can greatly improve the model performance, and that the prediction results are conducive to exploring dropout rates or online learning habits and performances.

1. Introduction

In traditional classroom teaching, a great number of learning interactions occur between students and instructors, between students and teaching materials, and between students and students. For instance, the students answer questions, read the textbooks, and take part in group activities. A good instructor can provide appropriate feedback and timely adjust the following interactions according to the typical patterns of behaviours shown by students in real-time (Hirumi, 2002). However, in the environment of online learning or E-learning, communication is technology-mediated and asynchronous, and the content and procedure of online courses are relatively fixed. There are limited opportunities for instructors to conduct individualized teaching and interpret information based on spontaneous responses, which necessitates the analysis and sequence of E-learning interactions. The course design and the arrangement of key interactions greatly affect learners' learning attitude and performance when studying online courses (Hirumi, 2002).

Wallace (2003) summarized that there are mainly four types of interactions that constitute students' engagement during online learning. Except for the common learner-instructor interactions, learner-content interactions, and learner-learner interactions, it also includes learner-interface interactions. As a matter of fact, the emergence of online education platforms makes it possible to capture and save massive data on users' learning interactions and then facilitates interaction analysis and design.

Massive Open Online Courses (MOOCs) (https://www.mooc.org) is one of the biggest online education platforms and provides free online courses for anyone to enroll in. It has attracted a tremendous number of users worldwide in the last two decades. Every day, millions of people log in to this platform and study courses of different subjects. During this process, a huge amount of data on learning interactions taken by users are generated (Dalipi et al., 2018). Every single operation that users carry out on https://www.mooc.org is recorded by MOOCs with the timestamp when that action occurs. Such interaction may be browsing a course, watching a lecture recording, joining a forum discussion, or even exiting MOOCs. Thus, for each user, there exists a chronological sequence of their interactions in the MOOC system (referred to as sequential interaction data or temporal interaction data).

As mentioned before, E-learning interaction analysis is of great significance for sequencing learning interactions and designing new courses. Furthermore, it helps educators gain a comprehensive understanding of users' online learning habits, which is especially useful to predict students' learning performances and provides reference when educators adjust course delivery styles and assessment methods (Moore & Blackmon, 2022). On the other hand, if the MOOC system can recognize a user's next action in advance, useful recommendations can be

* Corresponding author.
E-mail addresses: jr721@kent.ac.uk (J. Ren), s.m.wu@kent.ac.uk (S. Wu).

https://doi.org/10.1016/j.caeai.2023.100133
Received 16 November 2022; Received in revised form 10 March 2023; Accepted 13 March 2023
Available online 16 March 2023

made and shown directly to users, which makes learning more convenient and boosts the users' learning experience (Moore & Blackmon, 2022).

For the predictive analysis based on the sequential interaction data collected from MOOCs, the prediction of course completion or dropout rates has been a main concern for educators and researchers for years (Hone & Said, 2016, Dalipi et al., 2018, Goopio & Cheung, 2021). Nevertheless, Moore and Blackmon (2022) pointed out that researchers should pay more attention to the learner's learning action itself rather than only focusing on the overall MOOC dropout rates. The studies on prediction of interactions with MOOCs are not sufficient. More importantly, every interaction record captured by the MOOC platform shows "which user takes *which action* at *what time*" (Kumar et al., 2019). Thus, such temporal interaction data have important information on the action occurring time, but the temporal information hasn't been fully emphasized in the existing literature.

Thus, the following research questions are raised, and we aim to answer them in this paper.

- **Q1**: Considering both of temporal and sequential characteristics, whether the users' learning habits/patterns can be explored more comprehensively?
- **Q2**: Does the temporal information of interaction sequences plays an important role in the prediction of online learning interactions? If so, what kind of models can be used to describe interaction sequences with temporal information? And how can we simultaneously predict the user's next action and its occurrence time point when conducting the interaction prediction?

To solve the above questions, the related work on learners' sequential actions/interactions is summarised in the subsequent subsection.

1.1. Sequential action and interaction analysis

Analysing users' learning interactions have gradually aroused the interest of researchers in the past few years. Using statistical methods, Simpson et al. (2000) coded the behaviour of an elementary student and conducted a cause-and-effect analysis based on several sequences of interactions using general systems theory. Brodahl et al. (2022) investigated students' temporal interactions by conducting conversation analysis in the field of patient education. With machine/deep learning algorithms, Tang et al. (2016) applied the long short-term memory (LSTM) network to mine the underlying patterns of MOOC clickstream data. Sequential interaction rule mining was then studied by Fatahi et al. (2018), Lien et al. (2020) and Aktaş and Aktaş (2021) via sequential pattern mining algorithms. They identified the most frequent repetitive action sequences, generated feasible associate rules, and predicted student's learning style. Salehi (2013) produced a more effective recommendation by capturing users' sequential patterns compared with only making use of users' rating data. Most of these research works mentioned above merely focused on analysing users' interactions or making prediction by mining sequential patterns/rules without considering temporal information.

For temporal interaction analysis, Martínez et al. (2008) considered a temporal interactive effect to investigate the influence factors on reading performance and then predict reading achievement. Xia and Qi (2022) captured semantic features of learning behaviour taking the temporal risk and temporal tracking into account. Li et al. (2020) predicted the frequency of self-regulated learning behaviour occurring at time t via developing a multilevel vector autoregression model. Boroujeni and Dillenbourg (2018) focused on MOOC interaction sequences during the assessment period and proposed a temporal clustering pipeline to track learning interactions at each time step. Rajendran et al. (2018) proposed to use Hidden Markov models to represent and analyse the temporal learning interaction sequences.

It can be seen that few papers have fully taken the temporal characteristic into consideration. Vector autoregression method, clustering algorithm and Hidden Markov models have been used to describe the temporal sequence of learners' interactions with online platforms. However, these existing methods have the disadvantages of poor generalizability or relatively arduous calculation (Li et al., 2020, Boroujeni & Dillenbourg, 2018, Rajendran et al., 2018). Besides, there is little predictive work conducted from the perspective of considering the correlation between an interaction and the occurrence time of that interaction, and then predicting both at the same time. Thus, using powerful machine/deep learning methods to model and predict user temporal interactions with MOOCs is an interesting research direction for further investigation.

1.2. Marked neural temporal point process

The target of future temporal interaction prediction is to simultaneously predict the user action and the corresponding occurrence time, and the complexity of this kind of prediction makes it different from traditional time series forecasting. The differences mainly lie in two aspects: firstly, the actions from users do not necessarily occur at equal intervals, while the time series data are usually collected every equal time interval (such as every week or every month); secondly, parametric time series forecasting methods usually aim to predict one single index (such as the temperature, the stock price, or the product demand). However, in the case of temporal interaction prediction, we focus on *which action* a user will take at *what time*, and thus there are two parts involved–Action prediction and timestamp prediction, where the prediction of continuous occurrence time points is a regression problem, and that of actions belongs to a classification problem. Besides, existing recurrent neural networks (RNN) are usually applied to deal with one specific problem (namely, regression or classification or others), which is also inapplicable in the setting of this paper.

On the other hand, Vista et al. (2016) regarded sequential actions as markers and visualized a person's action sequence using directed graphs. Inspired by that, if we intend to predict user temporal interactions with MOOCs and the occurrence time points, the temporal point process (TPP) and the marked temporal point process (MTPP) can be applied.

The TPP is often used to model the data that are localized in a finite set of time points. Widely studied point processes include the Hawkes process (Hawkes, 1971), the Poisson process (Kingman, 1992) and the self-correcting process (Isham & Westcott, 1979). The MTPP considers the event marker of each time point in a TPP. Some basic concepts are briefly explained in Section 2.

Moreover, Du et al. (2016) first proposed the marked neural TPP (or neural MTPP), innovatively integrating neural networks and marked point processes as a non-parametric method for temporal prediction. The authors embedded the history of event occurrences into a compact vector and input both the marker and the interval duration into a recurrent layer. They then applied their proposed model to several real cases, including taxi operations, financial transactions, and electrical medical records, and achieved excellent prediction performances. Subsequently, a series of variants of neural MTPP or neural TPP models are introduced successively (see Xiao et al., 2017, Mei & Eisner, 2017, Omi et al., 2019, Shchur et al., 2019, Enguehard et al., 2020, Ben Taieb, 2022). The reader is referred to Shchur et al. (2021) for a comprehensive review of neural TPPs.

1.3. Novelty and contributions

In summary, the prediction of users' MOOC interactions integrated with temporal information has not been fully studied. In the existing papers on interaction prediction, repetitive sequence mining is one of the mainstream practices. Combining with temporal information, vector autoregression method, clustering algorithm and Hidden Markov models were proposed to analyse and then predict the learning interactions (see Section 1.1). Feasible deep learning methods are necessary to be put forward. On the other side, the marked neural TPP model focuses on the point process with event markers and has been successfully applied to the fields of finance and medical care (Du et al., 2016). Therefore, this paper aims to introduce the marked neural TPP model to predict user interactions with MOOCs based on temporal data, that is, predict the action and its occurrence time simultaneously.

The contributions of this paper include:

- An elaborate exploratory data analysis on the MOOC interaction dataset is given, and insights on users' typical patterns of learning behaviour are obtained;
- The concept of the MTPP is introduced to the research field of online education to temporal interaction data. Deep learning methods, including the LSTM and two marked neural TPP models, are tailored and then applied to predict the user's next action and its occurrence time on the MOOC platform;
- The prediction performances of the three different models are compared. The result shows that the marked MTPP models are efficient and accurate. Different hyper-parameter schemes are investigated as well;
- This work confirms that the correlation between actions and occurrence time points is extremely important in the prediction task, which enlightens future work to pay more attention to temporal information or multiple interlinked factors.

The remainder of this paper is organized as follows. Section 2 explains some basic concepts of the TPP. Section 3 introduces the methodologies, including the LSTM and the marked neural TPP, and prediction evaluation metrics. Section 4 describes and analyses the collected MOOC interaction dataset. Section 5 presents the model implementation and results. Section 6 summarizes the key findings and interprets the prediction results in the real application. Section 7 concludes our work and demonstrates limitations and future work.

2. Basic concepts

2.1. Temporal point process

The temporal point process (TPP) is a stochastic process composed of a series of time points of discrete events (Daley & Vere-Jones, 2008). Let $T_1, T_2, \cdots, T_n$ denote a sequential occurrence time points with $T_1 < T_2 < \cdots < T_n$. $T_1 = t_1, T_2 = t_2, \cdots, T_n = t_n$ is a realization of a TPP sequence. So, variables $X_i = T_i - T_{i-1}, i = 1, \cdots, n$ stand for the inter-event time intervals or waiting times between event occurrences, where $T_0 = 0$.

The counting process is defined by $N(t) = \sup\{n : T_n \le t\}$, which is the total number of events that have occurred before time point t.

In a TPP, the history of event occurrences may affect its re-occurrence in the future to some extent. Thus, as an important concept in the field of the TPP, the conditional intensity function is defined based on $N(t)$:

$$\lambda^*(t) = \lambda(t|H_{t^-}) = lim_{\Delta t \to 0} \frac{E(N(t, t+\Delta t)|H_{t^-})}{\Delta t},$$

where H_{t^-} stands for the occurrence history prior to time point t, and $N(t, t + \Delta t)$ is the total number of events occurring in the time interval $(t, t + \Delta t]$. Thus, the conditional intensity function can be interpreted as the chance that events are expected to occur around t, given the history up to that time point.

For example, the Hawkes process (Hawkes, 1971) is a counting process with the conditional intensity function

$$\lambda^*(t) = \mu + \alpha \sum_{t_j < t} \delta(t, t_j),$$

where $\delta(t, t_j) \ge 0$ is the trigger kernel function and $\mu \ge 0$ is the baseline intensity. The intensity function would increase by a certain amount when an event occurs in this case.

2.2. Marked temporal point process

For the MOOC temporal interaction prediction, if we only focus on one single action and regard it as the occurrence of an event, there is a TPP sequence for each user. For example, for user k, the TPP sequence is $U^k_{TPP} = (t^k_1, t^k_2, \cdots)$, where t^k_i stands for the occurrence time of that specific action. However, the user can conduct various actions on the MOOC platform, as mentioned before. Thus, if we take M different types of actions into account, the MTPP sequence for user k can be denoted by

$$U^k = \{(t^k_1, m^k_1), (t^k_2, m^k_2), \cdots, (t^k_{k_i}, m^k_{k_i})\},$$

where t^k_i is the i-th occurrence time point for user k and $m^k_i \in \{0, 1, \cdots, M-1\}$ is the marker of the occurring action. Note that t^k_i and m^k_i come in pairs ($i = 1, 2, \cdots, k_i$). And k_i is the length of time/action sequence of user k.

Furthermore, if the data are collected from N users in total, the whole dataset is a set of MTPP sequences $D = \{U^1, U^2, \cdots, U^N\}$ (Du et al., 2016).

The MTPP was originally applied to model earthquakes and aftershocks (Hawkes, 1971, Ogata, 1998). In recent years, it has been widely studied for analysing multiple events with the underlying spreading processes, such as rumours detection for social media (Naumzik & Feuerriegel, 2022) and recognition of the severity of illness on electronic health records (Islam et al., 2017).

Even though the traditional MTPP method has extensive applications, its weakness is also obvious. A parametric form of the point process is usually assumed first and then parameter estimation is carried out via some statistical methods. Thus, the choice of the pre-defined parametric form is extremely important. Nevertheless, the possibility that an appropriate model is chosen is very slim in real applications. Even for the well-performed model, it is not always reliable to project it to other situations (Du et al., 2016). That is the exact motivation of Du et al. (2016) to propose the non-parametric or semi-parametric marked neural TPP model. Neural networks such as RNN can be used to explore and form the underlying relationship by the model itself. Therefore, the marked neural TPP model will be applied to the temporal learning interaction prediction in this paper.

3. Methodologies

As mentioned above, the temporal features of action sequences can be captured by TPP models. If we consider several event types in one TPP sequence, an MTPP model is applicable. An MTPP regards the event type as a marker and then records the event occurrence time and the marker in pairs. Furthermore, the marked neural TPP, combining the LSTM and the MTPP, can simultaneously model user action and time point sequences.

Besides, LSTM networks have been widely applied because of their practicality in dealing with the prediction of time series data. Naturally, the time point and action prediction can be viewed as two separate data sequences, and then an LSTM network can be used to address the regression (for time points) and classification (for actions) problems, respectively.

3.1. Long short-term memory

The LSTM is one of the most popular variants of the RNN, which was introduced to overcome the challenge of long-term dependencies (Hochreiter & Schmidhuber, 1997). An LSTM has a cell state being passed straight through the neural network in different time steps, which stores useful long-term memories. It has three types of gate layers, and two of them are used to update the cell state, and then the obtained cell state and the input variable processed by another gate layer are integrated to update the hidden state. The calculation formulas of these gate layers are given in the following.

Forget Gate: The forget gate is used to forget the cell state whose information is obsolete. The activation function is a sigmoid function.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f).$$

where h_{t-1} is the hidden state of the previous time step and x_t is the input of the current time step.

Input Gate: The input gate is the second sigmoid layer, which is integrated with the output of a tanh layer to add new information to the cell state. The output result of that tanh layer is a new candidate cell state, denoted by $C_t^{\sim}$.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), C_t^{\sim} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C).$$

The final updated cell state C_t is obtained by

$$C_t = f_t * C_{t-1} + i_t * C_t^{\sim},$$

where f_t is derived in the forget gate.

Output Gate: The hidden state of the last time step and the input variable of the current time step would then go through the output gate, which is also a sigmoid layer.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o).$$

And the updated cell state obtained from the input gate is put through a tanh layer. Finally, by multiplying these two parts, the updated hidden state is output.

$$h_t = o_t * \tanh(C_t).$$

3.2. Marked neural temporal point process

To solve the problem of predicting the event marker and its occurrence time, Du et al. (2016) proposed the Recurrent Marked Temporal Point Process (RMTPP) and Xiao et al. (2017) introduced the Event Recurrent Point Process (ERPP), which are the two earliest marked neural TPP models. Thus, these two models will be considered to conduct the MOOC temporal interaction data analysis. Here, the rationales of the RMTPP and the ERPP are explained as follows.

In Section 2.2, we state that the interaction dataset has the following form

$$D = \{U^1, U^2, \cdots, U^N\},$$

where $U^k = \{(t_1^k, m_1^k), (t_2^k, m_2^k), \cdots, (t_{k_i}^k, m_{k_i}^k)\}$ are the data for the user k, and (t_i^k, m_i^k) is one temporal interaction in pair, and t_i^k is the occurrence time point and m_i^k is the action marker ($i = 1, 2, \cdots, k_i$).

The conditional intensity function of a TPP is denoted by $\lambda^*(t)$. The basic idea behind a marked neural TPP is to not directly define an exact parametric form for $\lambda^*(t)$. On the contrary, through the information propagation and iteration in the neural networks, an intensity function having a non-parametric or semi-parametric form can be derived.

After getting the conditional intensity function, the conditional density function can be obtained as follows.

$$f^*(t) = \lambda^*(t) \exp\left(-\int_{t_n}^{t} \lambda^*(\zeta) d\zeta\right). \quad (1)$$

The loss function is defined as the negative joint log-likelihood function. Utilizing the conditional density function $f^*(t)$, the loss function and the predicted next time point can be calculated by

$$l = -\sum_k \sum_i \left(\log\left(P(m_{i+1}^k | h_i)\right) + \log\left(f(d_{i+1}^k | h_i)\right)\right),$$

and

$$\hat{t}_{j+1} = \int_{t_j}^{\infty} t \cdot f^*(t) dt,$$

where d_{i+1}^k is the inter-event time. The inter-event time is the final form for the time point sequence to be input into the recurrent layer.

The modelling structures of RMTPP and ERPP algorithms are given in Figure 3 of Du et al. (2016) and Figure 2 of Xiao et al. (2017), respectively. In the RMTPP model, the event marker is put through an embedding layer and then combined with time information. They are input together to the LSTM layer subsequently. For the ERPP algorithm, there are two LSTM networks corresponding to time and event sequences, respectively. The outputs of the two LSTMs are integrated into an embedding mapping layer and then used to carry out the prediction.

For the RMTPP model, the conditional intensity function $\lambda^*(t)$ is defined with a partially parametric form as follows.

$$\lambda^*(t) = \exp\left(\boldsymbol{v}^{t^T} \cdot \boldsymbol{h}_j + w^t(t - t_j) + b^t\right).$$

Then, substituting the conditional intensity function $\lambda^*(t)$ to Equation (1), the conditional density function $f^*(t)$ can be written as

$$f^*(t) = \exp\Big\{\boldsymbol{v}^{t^T} \cdot \boldsymbol{h}_j + w^t(t - t_j) + b^t + \frac{1}{w}\Big(\exp(\boldsymbol{v}^{t^T} \cdot \boldsymbol{h}_j + b^t) - \exp(\boldsymbol{v}^{t^T} \cdot \boldsymbol{h}_j + w^t(t - t_j) + b^t)\Big)\Big\}.$$

For the ERPP model, it directly defines the loss function of time prediction as a square loss. The loss function of event prediction is the cross-entropy loss function, which is defined as

$$E = -\sum_j y_j \log(z_j),$$

where z_j is the probability vector output from the classification model and y_j is the real class with one-hot coding. Thus, the total prediction loss is the sum of the loss for time prediction and the cross-entropy loss for the event.

3.3. Evaluation metrics

In the temporal interaction prediction, the time point prediction and the event prediction are a regression task and a classification task, respectively. Thus, different evaluation measures are supposed to be applied to them.

For regression problems, some commonly-used evaluation metrics include the Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error. In this paper, we will use MAE to evaluate the performance of time point prediction, which can be calculated by

$$MAE = \frac{1}{n \times k_i} \sum_k \sum_{i=1}^{k_i} |t_i^k - \hat{t}_i^k|,$$

where n is the total number of users included in prediction, and t_i^k is the real time point and $\hat{t}_i^k$ is the predicted value.

For the action classification, the precision, the recall and the F1 value (Macro-average method for multi-class classification) can all be used to evaluate the performance of the models. The definitions of the performance metrics are given by

$$Precision = \frac{1}{\sum_{m=0}^{M-1} I_{\#(\hat{m}_i^k = m) \neq 0}} \sum_{m=0}^{M-1} \frac{\#(\hat{m}_i^k = m_i^k = m)}{\#(\hat{m}_i^k = m)}, \quad \#(\hat{m}_i^k = m) \neq 0,$$

and

$$Recall = \frac{1}{\sum_{m=0}^{M-1} I_{\#(m_i^k = m) \neq 0}} \sum_{m=0}^{M-1} \frac{\#(\hat{m}_i^k = m_i^k = m)}{\#(m_i^k = m)}, \quad \#(m_i^k = m) \neq 0,$$

Table 1
The description of the MOOC interaction dataset.

Variable	Type	Description	Value
id	(int)	The user id	0-7,046
time	Scale (float)	The time point when the user took an action	Start from timestamp 0 (Unit: 10^4 seconds)
action	Nominal (int)	The corresponding action taken by user	0-96

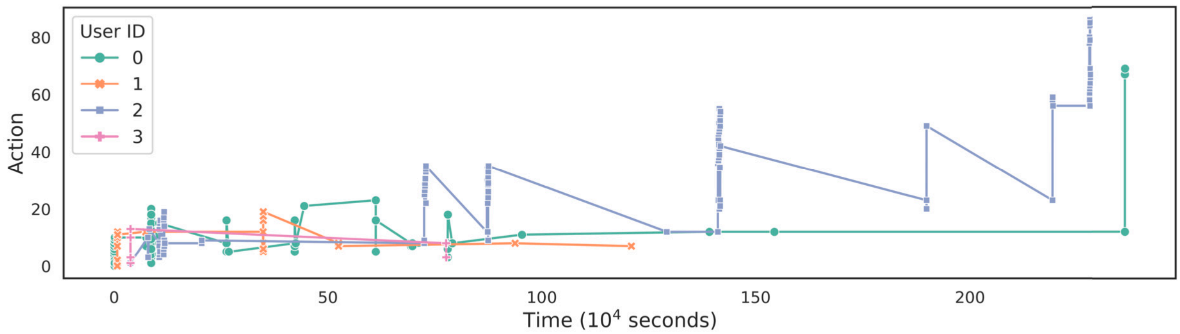


Fig. 1. The action sequences of user 0-3 in the MOOC interaction dataset.

where $m_i^k \in \{0, 1, \cdots, M-1\}$ is the actual marker of the i-th occurring action for user k, $\hat{m}_i^k$ is the predicted action, $\#(A)$ stands for the total number of cases satisfying the criterion A, and I_A is an indicator function where $I_A = 1$ if A is True and $I_A = 0$ otherwise.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

4. Data collection and processing

4.1. MOOC interaction dataset

The MOOC interaction dataset we will analyse is a processed version from Kumar et al. (2019). The original dataset is open-access and was used in the Knowledge Discovery and Data Mining (KDD-CUP) competition of the year 2015. Kumar et al. (2019) collected and processed the dataset for social networking study. The specific information of this dataset is provided in SNAP (2019).

The dataset focuses on 97 different user actions from 7,047 users of the MOOC platform. The data was collected across one month, and the final dataset has 411,749 temporal interactions in total. The involved user actions include doing homework, watching videos, reading the Wikipedia of a course, discussing in the forum, browsing the content of courses, and finishing the quiz, etc.

There are 3 variables in this MOOC interaction dataset, namely, id, time, and action. For privacy protection, the users and the taken actions are coded by numbers starting from 0, and the occurrence time is also standardized as beginning from time point 0. The concrete description of the variables is given in Table 1.

The action sequences of the first 4 users in this dataset are plotted in Fig. 1 as examples. It can be observed that the sequence length varies from person to person; the user's learning pattern is usually to conduct a series of actions during a very short period and then stays still for a period, and then continue to conduct a bunch of actions; and as time goes on, the time interval between clustered actions has a gradually increasing trend.

4.2. Exploratory analysis

This section is devoted to the exploratory analysis of this interaction dataset. The descriptive statistics of the 3 variables are presented in Table 2, including the mean value (Mean), the standard deviation (Std), the minimum/maximum value (Min/Max), the quartiles (25%, 50%, 75%) and the count number (Count). It can be seen from Table 2 that the 411,749 temporal interactions from 7,047 MOOC users are recorded properly. There is neither missing value in this dataset nor other type of dirty data such as inconsistent data. Thus, there is no need for additional data cleaning.

Table 2
The descriptive statistics of variables in the MOOC interaction dataset.

Variable	Mean	Std	Min	Max	(25%, 50%, 75%)	Count
id			0	7,046		411,749
time	138.3	73.9	0	257.2	(78.5, 145.8, 201.2)	411,749
action	26	21.2	0	96	(8, 22, 38)	411,749

Table 3
The top 5 and last 5 users ordered by the length of action sequences.

Index	Last		Top	
	User ID	Sequence length	User ID	Sequence length
1	1673	5	1181	505
2	1650	5	1686	470
3	6559	5	805	463
4	2690	5	2990	439
5	4339	5	2358	406

To better understand the learning actions of these 7,047 MOOC users during that one month, the length of the respective action sequence of each user (namely, the total number of actions taken by each user) is investigated. Table 3 lists the length of the 5 longest and 5 shortest action sequences and their corresponding users. The most active learner is the one with id number 1181, and he/she conducted 505 actions in one month; however, the laziest users have only operated five times on the MOOC website.

Fig. 2 is the box plot for the length of action sequences of the whole user community in this dataset. The median is 38 and the mean value is 58. Most of the users have action sequences with lengths less than 100 and only a small part of them took more than 200 actions.

From the perspective of 97 different actions, the count numbers of each action are shown in Fig. 3 and the detailed information of the top/last 5 most frequent actions is presented in Table 4. It can be found that the actions with smaller id numbers are performed more frequently, but different actions are relatively evenly-distributed because the most frequent action taken by users only accounts for 4.7% of the total interactions.

Because the time interval between actions is an important concept in the research field of the TPP, here, we visualize the approximate

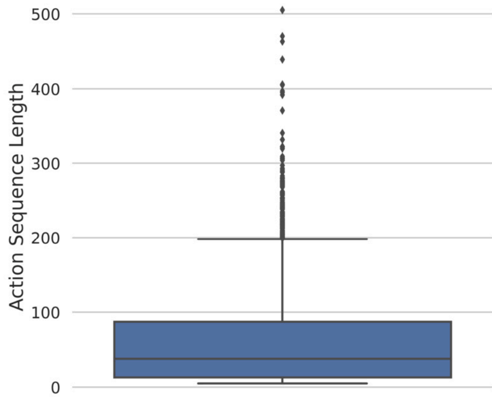


Fig. 2. The length of user action sequences in the MOOC interaction dataset.

Table 4
The top 5 and last 5 actions ordered by the frequency of actions.

Index	Last		Top		
	Action ID	Total times	Action ID	Total times	Percent (%)
1	93	87	8	19,474	4.7
2	92	95	7	16,352	4.0
3	94	107	21	16,182	3.9
4	89	122	9	14,709	3.6
5	90	128	3	14,566	3.5

distributions of the inter-action time from 3 users (see Fig. 4). Note that the unit of time here is second.

Fig. 4 shows that the density distributions of the time interval from different MOOC users have relatively similar shapes and trends; the distribution is relatively concentrated (i.e., the kurtosis is large) and is skewed to the left, and most of the inter-action intervals are distributed in the range of 0-50 seconds except for some extreme points, and the interval length is very close to 0, indicating that most actions were taken successively and followed by the next action quickly.

4.3. Data processing

In this subsection, the MOOC interaction dataset will be processed to satisfy the requirements of model inputs, mainly including two steps. The first one is to randomly split the whole dataset into 3 sub-datasets used for training (60%), test (20%) and validation (20%), respectively, where the training dataset will be used to train a prediction model, the validation dataset is used for checking the model convergence, and the test dataset is used for evaluating the performance of the model. The second step is to obtain time point sequences and action sequences with a pre-defined time step from these 3 sub-datasets.

The two steps of data processing are presented in Fig. 5.

Step 1: Due to the temporal correlation among interaction data of an individual, the records for each user cannot be split or reordered. Instead, we regard each individual's data as one user block and then split the entire dataset via splitting different user blocks. Through generating random numbers, the order of user blocks is shuffled, but the temporal order of the time and action sequences for every user block remains unchanged. The training, validation and test datasets are obtained according to the percentages set in advance (60%, 20% and 20%). Finally, there are 4,228 users in the training dataset, 1,409 users in the validation dataset and 1,410 users in the test dataset. The descriptive statistics of the training, validation and test datasets are summarized in Table 5. After random splitting, the distribution of each variable in three sub-datasets is relatively close to each other, and in this way the effectiveness and generality of the trained model can be guaranteed.

Table 5
The respective descriptive statistics for the training, validation and test datasets.

Variable	Time			Action		
	Training	Validation	Test	Training	Validation	Test
Mean	137.8	140.2	137.9	27	26	27
Std	74.4	72.9	72.9	21.1	21.4	21.3
Min	0.0	0.7	3.8	0	0	0
25%	77.5	80.6	78.6	8	8	8
50%	144.9	147.8	145.1	22	21	22
75%	201.3	201.7	199.6	38	38	39
Max	257.2	257.2	257.1	96	96	96
Count	248,656	78,881	84,212	248,656	78,881	84,212

Table 6
The descriptive statistics of the standardized time for the training, validation and test datasets.

Variable	Time		
	Training	Validation	Test
Mean	0.535	0.543	0.529
Std	0.289	0.284	0.288
Min	0.000	0.000	0.000
25%	0.301	0.311	0.295
50%	0.563	0.573	0.557
75%	0.782	0.783	0.772
Max	1.000	1.000	1.000
Count	248,656	78,881	84,212

Step 2: For each user block in each sub-dataset, we split the complete sequences of that user into several time and action sub-sequences with a specific time step. At each split, we move forward one unit to make full use of data information. For example, if we set the time step as 10, then the time and action sub-sequences are records with Index 1-10, 2-11, 3-12... (See Fig. 5). On the one hand, the length of a user's action and time point sequences varies from user to user, but the format of input data for the LSTM network requires us to specify the same time steps. On the other hand, the data created a long time ago is relatively obsolete for predicting the next step, because the closer the interaction, the greater the impact on the user's decision-making.

5. Experimental results

In this section, the methodologies mentioned in Section 3, namely the LSTM, RMTPP and ERPP models are constructed based on the MOOC interaction dataset. The action and time sequences obtained from Section 4.3 are fed into the models and Fig. 6 presents the flow charts. Here, for these three models, the structure of model layers and the dimensions of each layer are set to be close to each other to facilitate the model comparison, and the detailed description will be given in the following subsections.

5.1. Long short-term memory

5.1.1. Model implementation

Two LSTM networks are built for time and action sequences respectively. For the occurrence time prediction, the timestamps are first standardized to (0, 1) interval in the training, validation, and test datasets, respectively, to facilitate model training. The descriptive statistics of the standardized time for training, validation and test datasets are shown in Table 6. For the action prediction, one-hot coding is needed for each action in every interaction (See Fig. 6a). We consider the time step is equal to 10.

For the model structure, in the time prediction LSTM model, we set two LSTM layers, two Dropout layers and two Dense layers (see Table 7). The dropout layer is a regularization technique which can be utilized to prevent a neural network from overfitting. The input dimension is just 1, and then the neuron units in each layer are listed,

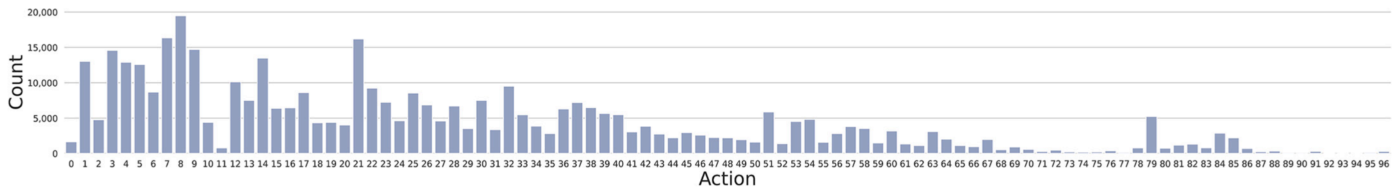


Fig. 3. The counts of 97 actions in the MOOC interaction dataset.

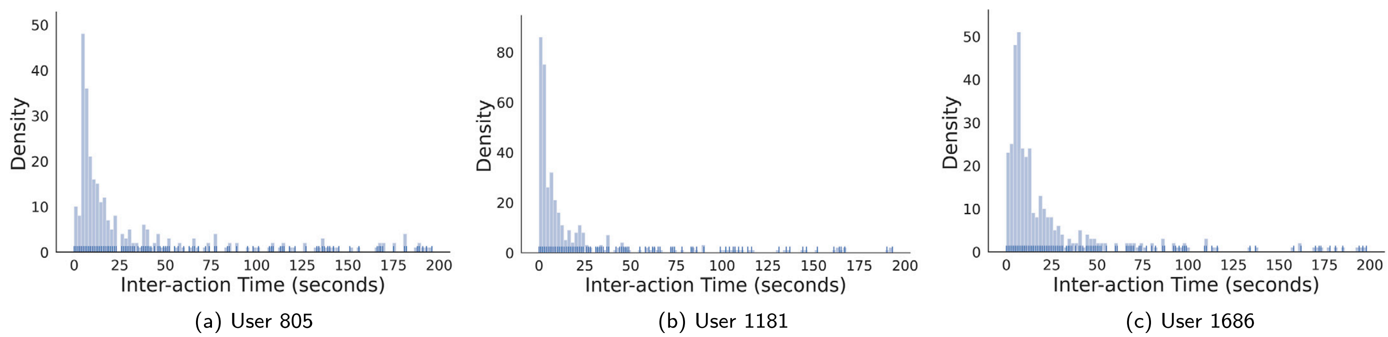


Fig. 4. The distributions of inter-action time for 3 users in the MOOC interaction dataset.

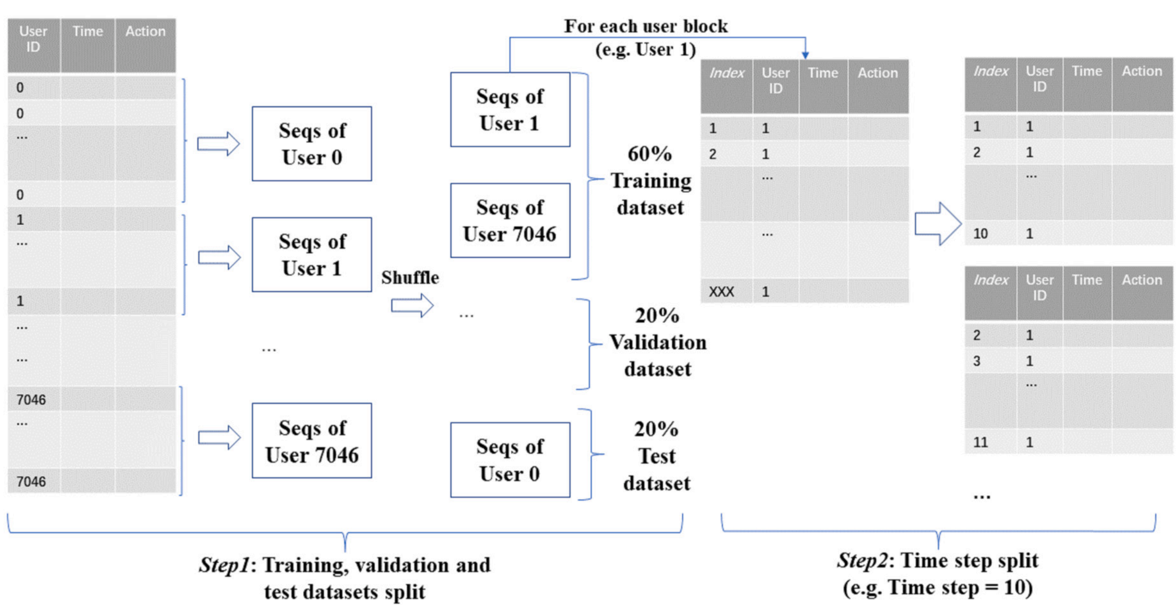


Fig. 5. Data processing flow chart of the MOOC interaction dataset.

Table 7
The LSTM model structure of time prediction.

Layer	Units	Number of parameters
LSTM	64	16,896
Dropout	64	
LSTM_1	128	98,816
Dropout_1	128	
Dense	64	8,256
Dense_1	1	65
	Total	124,033

Table 8
Training parameters for LSTM models.

Learning rate	Batch size	Training epoch
0.001	16	30

respectively. The activation function is the Relu function and the loss function is the mean square error. The constructed model is visualized in Fig. 7 and the training parameters are given in Table 8.

While for the LSTM model used for action prediction (see Table 9 and Fig. 8), given that the matrix obtained by action one-hot coding is high-dimensional and sparse, an embedding layer is added to make the information more compact and improve the efficiency of model training. Finally, two Dense layers output a predicted vector with 97 dimensions. Note that in the second Dense layer, the activation function is the Softmax function to transform the multi-class classification results. The loss function is the categorical cross-entropy function. Other training parameters are also listed in Table 8.

5.1.2. *Prediction results*

We apply the trained models to the sequences in the test dataset. The results of the occurrence time and action prediction are given as follows.

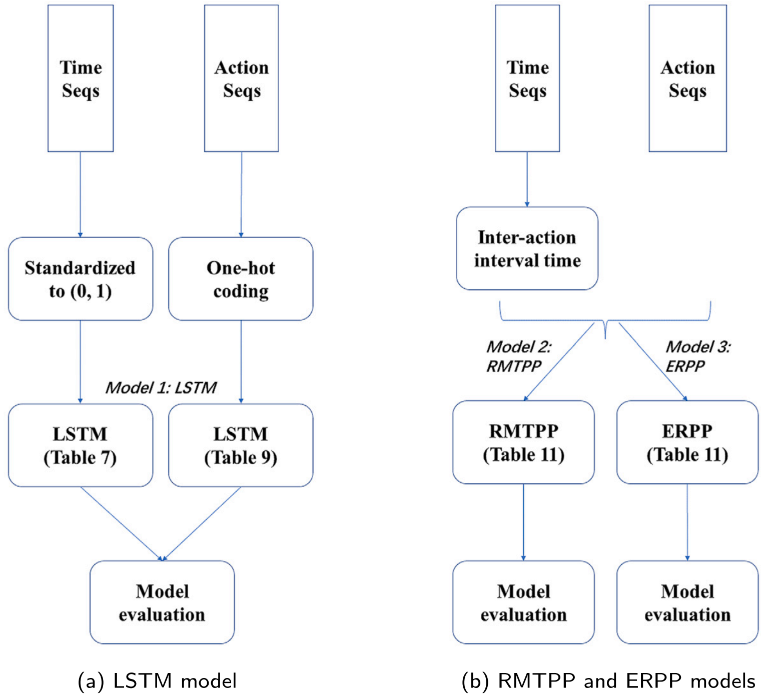


Fig. 6. The flow chart of model developing.

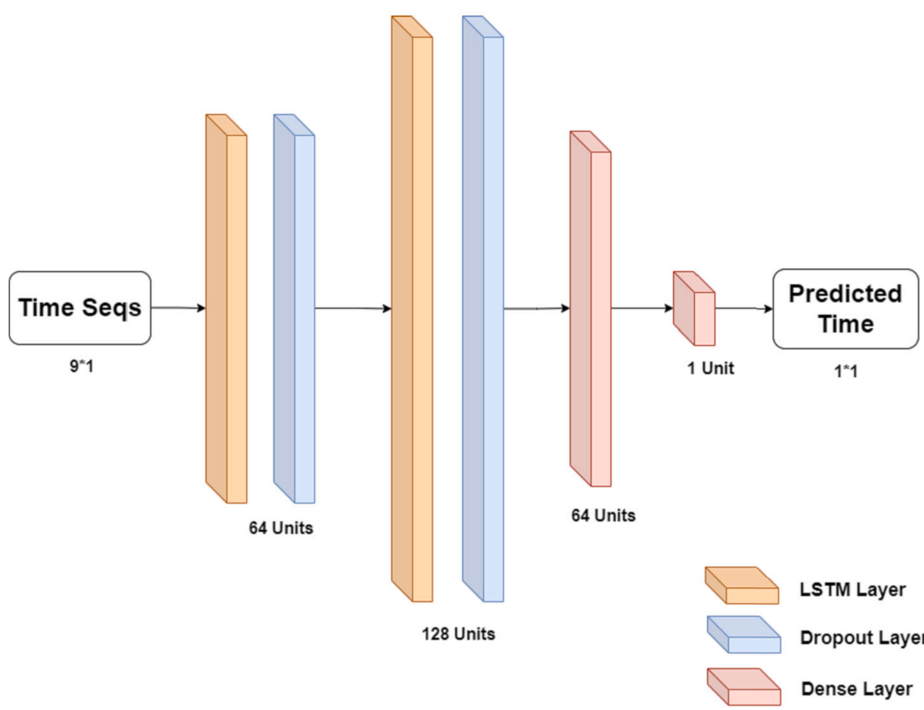


Fig. 7. The LSTM model structure of time prediction.

Time point prediction

It is revealed that the MAE between the predicted time point and the real occurrence time is 3.890 for the training dataset, and 3.814 for the test dataset. No over-fitting occurs.

To acquire an intuitive understanding of the prediction performance, the prediction results of the first 1,000 items in the test dataset are visualized in Fig. 9 and Fig. 10, respectively.

From Fig. 9, we can see that the first 1,000 values consist of multiple time sequences from several different users; the plotted lines of the predicted time and the real time are close to each other and have similar trends (increasing or nearly stable); the predicted time is slightly larger than the real time point in most cases.

Because the predicted value and the real time point are so close, although the predictive time point is larger on most occasions, the magnitude of their difference is too small to distinguish. It can be seen in Fig. 10, most of the scatter points distribute around the diagonal line, indicating that the model performs relatively well. However, there are a small part of points showing the predicted time point is far less than the real time. The reason is that the prediction has an obvious lag when there is a big jump for the time value in the sequence of every user.

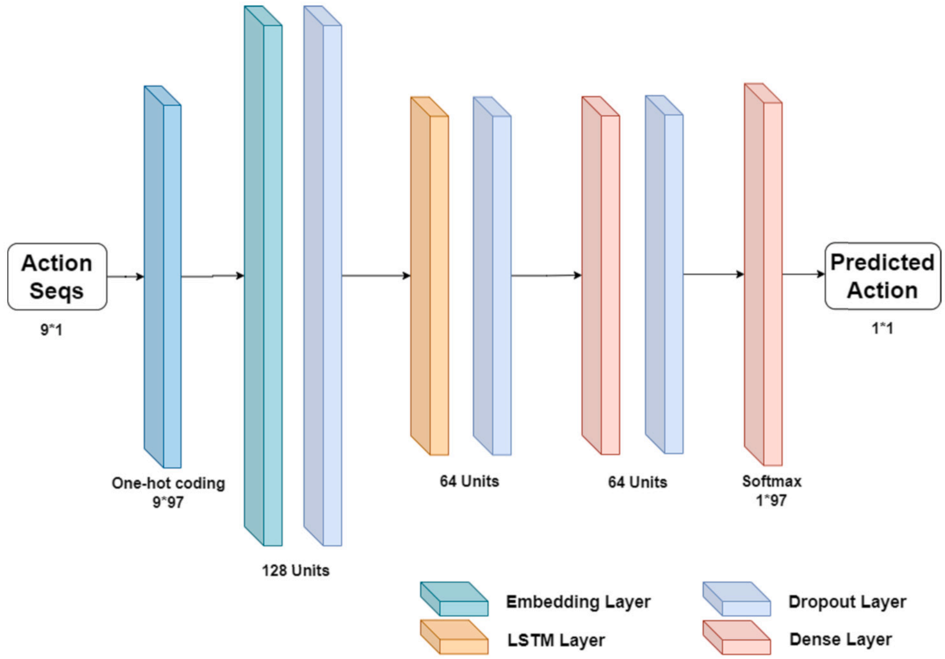


Fig. 8. The LSTM model structure of action prediction.

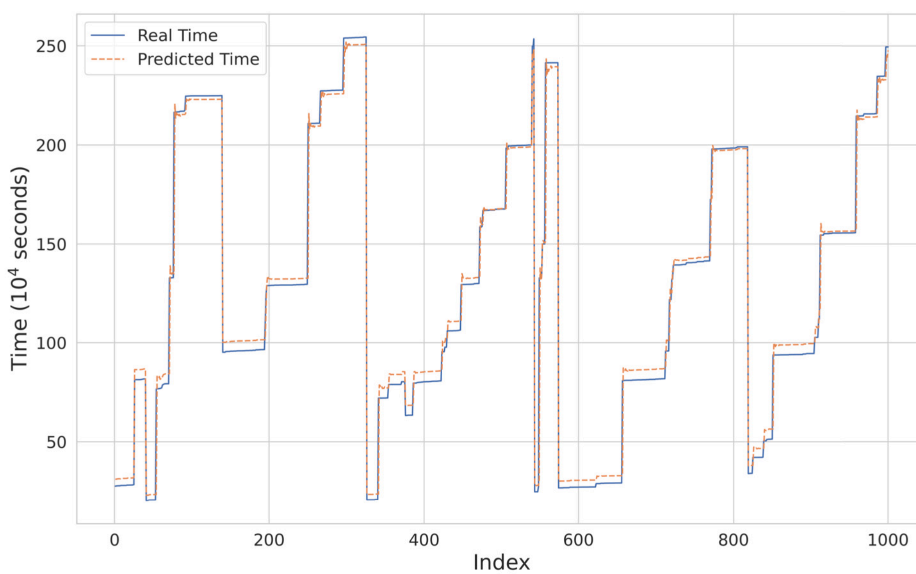


Fig. 9. The comparison of real time and predicted time for the first 1,000 items in the test dataset-Part I.

Table 9
The LSTM model structure of action prediction.

Layer	Output shape (Units)	Number of parameters
Embedding	(, 9, 128)	12,416
Dropout	(, 9, 128)	
LSTM	(, 64)	49,408
Dropout_1	(, 64)	
Dense	(, 64)	4,160
Dropout_2	(, 64)	
Dense_1	(, 97)	6,305
	Total	72,289

Action prediction

The precision, the recall and the F1 value of the model for action prediction on the training and test datasets are listed in Table 10. Both the precision and recall are about 50%, which is relatively high for multi-class classification tasks. The performance of the model on the training dataset is similar to that on the test dataset, which can be explained from two perspectives. Firstly, the training and test datasets were divided evenly. As shown in Table 5, the distribution of data in the training and test datasets is really close to each other. Secondly, the precision and recall values are defined for the multi-class classification (97 categories in total) based on the macro-average method. As shown in Section 3.3, the precision (or recall) is obtained as the average value of correctly classified percentage for each category. This method can be greatly affected by categories with relatively small sample sizes, and thus the improvement on the training dataset is not significant, compared with the test dataset.

The scatter plot of the first 300 actions and the heatmap for all the predicted actions in the test dataset are shown in Fig. 11 and Fig. 12, respectively. About half of the square and triangle scatter points coincide with each other. The heatmap presents the confusion matrix in a vivid

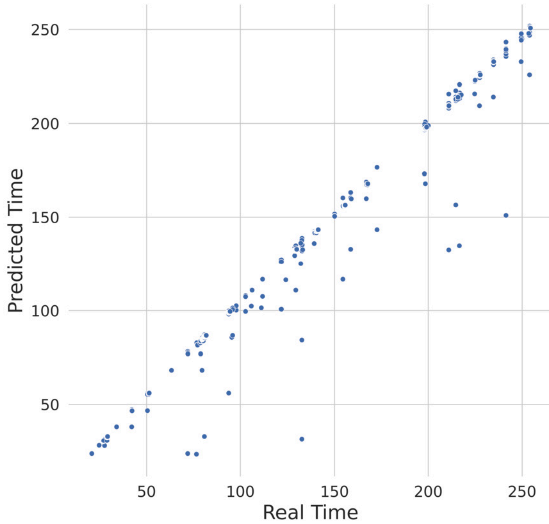


Fig. 10. The comparison of real time and predicted time for the first 1,000 items in the test dataset-Part II.

Table 10
The model performance of action prediction for the training and test datasets.

Dataset	Precision (%)	Recall (%)	F1 value (%)
Training	50.3	49.8	50.1
Test	49.6	48.8	49.2

way. In Fig. 12, the colour blocks in the diagonal line are relatively dark, and the prediction for the actions coded as the first 40 are more accurate due to their relatively larger sample size. The LSTM model can realize an effective multi-action classification for the MOOC interaction dataset to some extent.

5.2. Marked neural point process

5.2.1. Model implementation

The basic model structure for the RMTPP or ERPP algorithms is mentioned in Section 3.2. Specifically, the description of the recurrent layers is given in Table 11 and the visualization is presented in Fig. 13. Here the input time is the inter-action time interval instead of the occurrence time point (see Fig. 6b).

The structure shows that there is an Embedding layer for the input action, and then it is concatenated with the input time interval, which is then put through an LSTM layer to obtain a hidden state. Subsequently, three Dense layers are utilized to output the final prediction. The dimensions of the Embedding layer, hidden state and the first Dense layer are denoted by *emb_dim, hid_dim* and *mlp_dim* respectively in Table 11. Here, we set *emb_dim* = 128 and *mlp_dim* = 16, and let the model iterate 30 epochs. The sequence length, *hid_dim*, batch size and learning rate are the hyper-parameters we will discuss further.

The activation function is set to be the tanh function. The loss function is a function of time loss and event loss, and there is a multiplicative weight $\alpha = 0.05$ for time loss to balance the magnitude of these two parts of losses.

5.2.2. Prediction results

RMTPP model

To fully explore the RMTPP model performance, several experiments with different hyper-parameter schemes are conducted. The details of these hyper-parameter settings are shown in Table 12. The scheme with sequence length 10, *hid_dim* 64, batch size 16 and learning rate 0.001 (indexed as Scheme I) can be viewed as a baseline scheme, which is the same as in the LSTM models. And then, these hyper-parameters are set to be a series of different numbers in Scheme II-V. Table 12 highlights the differences between Scheme II-V and Scheme I, and gives the form of shorthand for each scheme in the last column. The model results will be observed under different scenarios.

In the training process of the RMTPP model, the loss values for the training dataset and the validation dataset in each iteration epoch are recorded. The results are presented in Fig. 14 (the subfigures on the left). At the end of every epoch, the trained model is evaluated on the validation dataset. The time error (namely MAE) for time prediction and the F1 value for action prediction are obtained. The iteration trends can be seen from the subfigures on the right of Fig. 14.

The trained models after 30 epochs are applied to the test dataset, and the evaluation results are summarized in Table 13.

From Fig. 14, some conclusions can be drawn:

(1) As the model iterates, the loss on the training dataset and that on the validation dataset decrease and finally tend to be stable. The F1 value gradually increases in the iteration, while the time error fluctuates but shows an upward trend at the beginning of the iteration and it also finally reaches a steady state.

(2) The iteration trends of the RMTPP model are slightly different under different hyper-parameter schemes.

(3) In Scheme II, we can see the loss on the validation dataset keeps almost unchanged at the end, indicating a model convergence; however, the loss on the training dataset is continuously decreasing, although the decreasing speed gradually slows down. The model is a little overfitting under the corresponding hyper-parameter setting. The hidden state with 128 dimensions may involve excessive parameters. The prediction results are not so good as well in this case (see the time error).

(4) If we try to add more information to the input data, namely increasing the sequence length and batch size, the subfigures (e) - (h) show the results. Increasing the batch size from 16 to 32 has almost no influence on the model performance, whether for the loss or the time error and the F1 value. While for sequence length, there is a slight improvement after it increases to 20. The time error is relatively steady in the iteration process.

(5) Compared with other schemes, the differences between the losses on the validation dataset and the training dataset in Scheme V are the smallest. In Scheme V with the learning rate of 0.0001, the model achieves the smallest error in predicting gap time; however, the action prediction doesn't perform as well as the other four.

Applying the models to the test dataset, the same conclusions can also be drawn from Table 13. The time prediction of the RMTPP model performs the best in Scheme V, and the action prediction achieves the best performance in Scheme III. But the best performance on either one hand is always at the expense of the poor performance on the other. For instance, the indices of action prediction for Scheme V are relatively low. Overall, the model performance for Scheme I is the most balanced, which can be regarded as the most appropriate hyper-parameter combination for the RMTPP model based on the MOOC interaction dataset.

ERPP model

In the ERPP model, the same hyper-parameter setting as in Scheme I is used to facilitate the subsequent model comparison. Similarly, Fig. 15 presents the iteration plots of training and validation losses as well as the evaluation on the validation dataset for every epoch.

From the loss curves, it can be observed that the ERPP model converges well, and its convergence speed is faster than that of the RMTPP model. In the iteration process, the performance of action prediction is boosted gradually, while the time error keeps fluctuating between the interval (2.7, 2.9).

The well-trained ERPP model is then applied to the test dataset, and the evaluation results are listed as follows.

Time error (MAE): 2.638; Precision: 50.0%; Recall: 46.2%; F1 value: 48.0%.

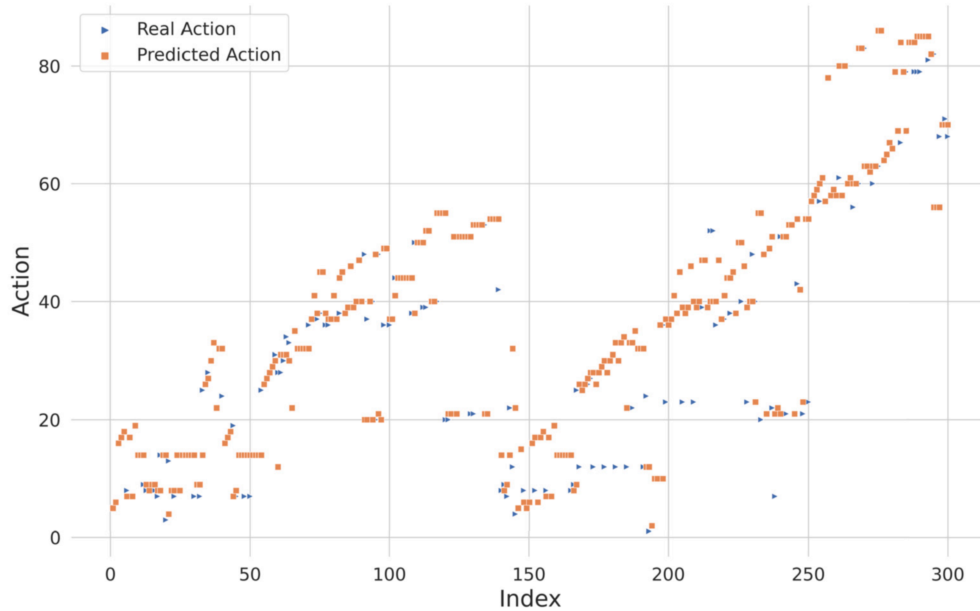


Fig. 11. The comparison of real action and predicted action for the first 300 items in the test dataset.

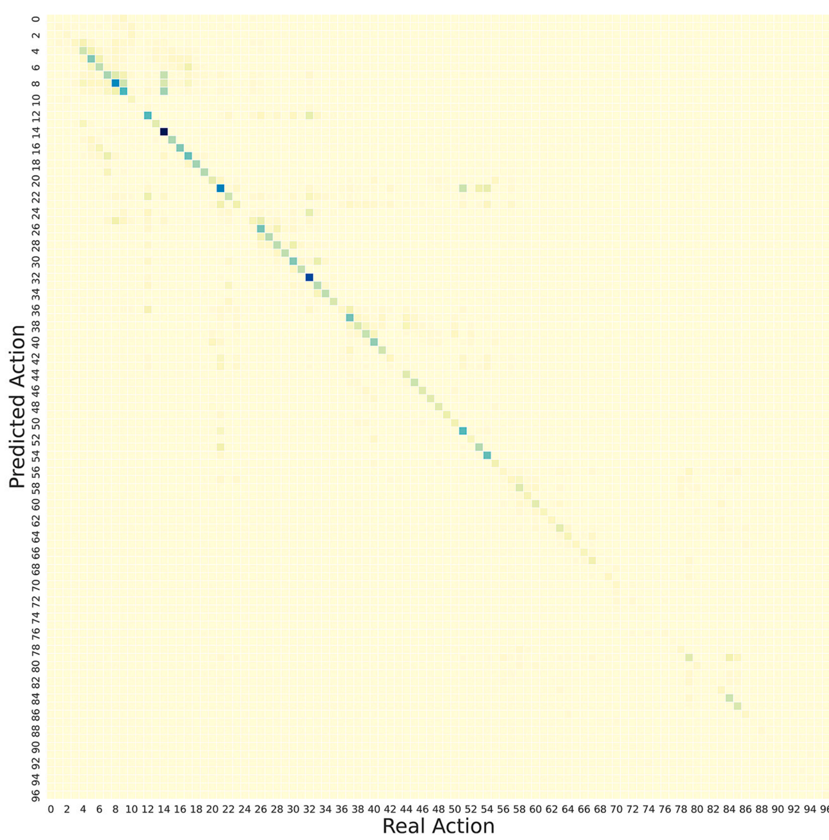


Fig. 12. The heatmap of the comparison between real action and predicted action.

5.3. Model comparison

Based on the MOOC interaction data, three different algorithms, namely the LSTM, RMTPP and ERPP models, are applied to predict the next user action and its occurrence time, respectively. With the identical model configuration parameters (that is, 10-64-16-0.001), the three models are implemented, and the prediction results are summarized in Table 14, including the time error (MAE) for time prediction as well as the precision, the recall and the F1 value for action prediction.

Overall, all these three models can capture the features of MOOC interaction and then conduct effective temporal predictions for the next learning action and the occurrence time point.

Compared with the pure LSTM model, the performance of temporal interaction prediction with the RMTPP and ERPP models greatly improves, and the time error decreases from 3.8 to 2.4 or 2.6 (Unit: 10^4 seconds). The LSTM model separately treats the time and action sequences, while the RMTPP and ERPP models describe them together using the marked neural TPP. Thus, the results indicate the importance

Table 11
The structure of the RMTPP and ERPP algorithms.

Layer	Input and output dimension	Description
Embedding	97 -> emb_dim	Action embedding layer
Dropout		
LSTM	emb_dim + 1 -> hid_dim	Concatenate the embedded action and the time point
Dense	hid_dim -> mlp_dim	
Dropout_1		
Dense_1	mlp_dim -> 97	The output for action prediction
Dense_2	mlp_dim -> 1	The output for time prediction

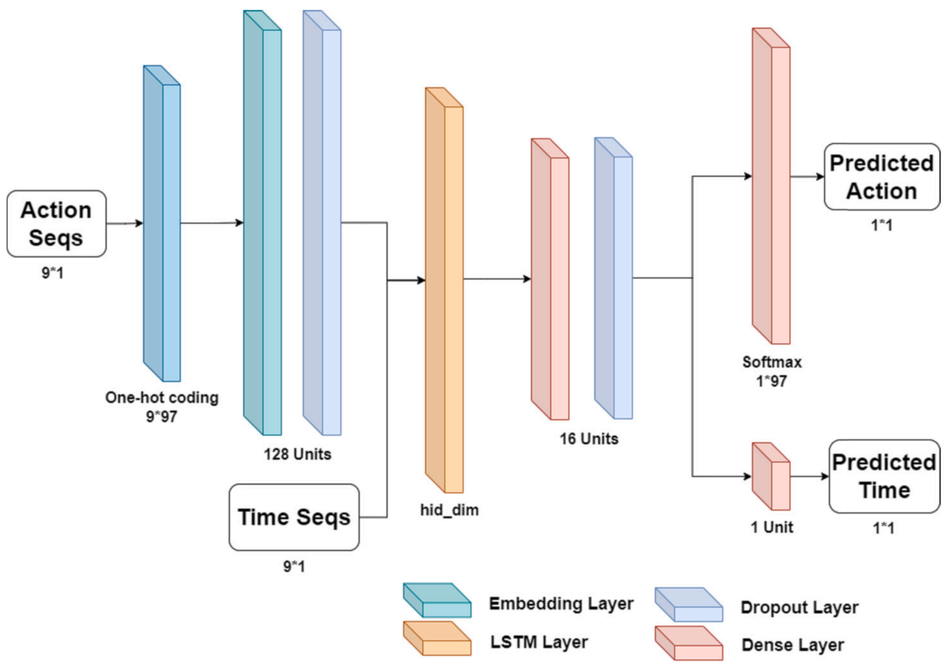


Fig. 13. The structure of the RMTPP and ERPP neural networks.

Table 12
The hyper-parameter combinations for the RMTPP model.

Scheme index	Sequence length	hid_dim	Batch size	Learning rate	Labels
I	10	64	16	0.001	10-64-16-0.001
II	10	**128**	16	0.001	10-128-16-0.001
III	**20**	64	16	0.001	20-64-16-0.001
IV	10	64	**32**	0.001	10-64-32-0.001
V	10	64	16	**0.0001**	10-64-16-0.0001

Table 13
The model evaluation for different hyper-parameter schemes based on the test dataset.

Scheme index	Time Error (MAE)	Precision (%)	Recall (%)	F1 value (%)
I	2.479	51.2	49.2	50.2
II	2.593	51.8	49.3	50.6
III	2.520	**53.7**	**49.4**	**51.4**
IV	2.507	51.0	49.1	50.0
V	**2.214**	49.0	42.5	45.5

Table 14
The MOOC interaction prediction of the LSTM, RMTPP and ERPP models.

Model	Time Error (MAE)	Precision (%)	Recall (%)	F1 value (%)
LSTM	3.814	49.6	48.8	49.2
RMTPP	2.479	51.2	49.2	50.2
ERPP	2.638	50.0	46.2	48.0

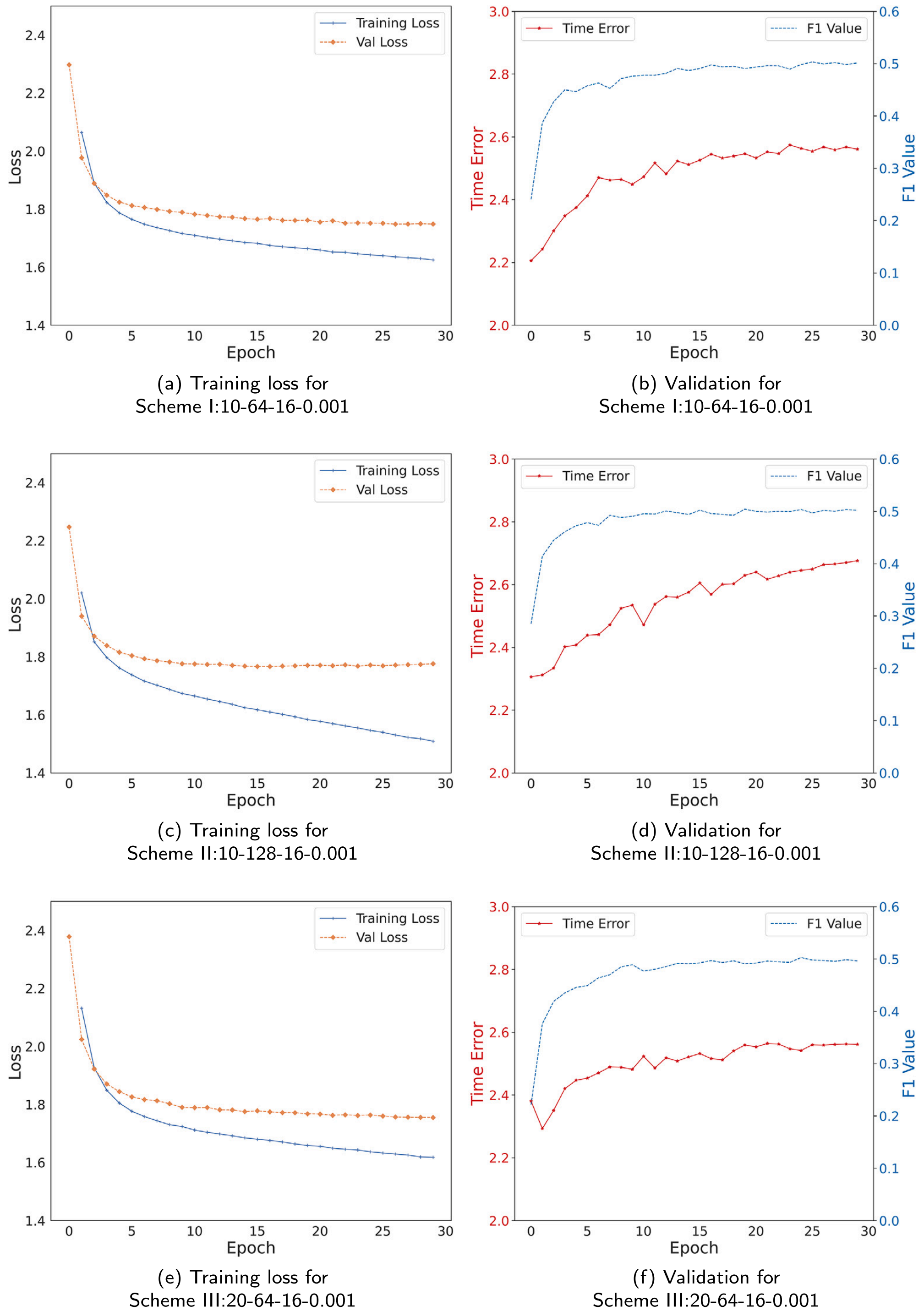
of considering the correlation between time and action when predicting user temporal interactions.

Note that the way of considering the correlation between two sequences can drastically reduce the error in predicting the gap time, but there is almost no difference for action prediction. In these three models, The ERPP model performs the worst in terms of the recall and the F1 value.

From each model, the time prediction of the LSTM model has an obvious lag when the user's inter-action time interval increases remarkably, which is one of the reasons for the relatively large time error of the LSTM model. And that is also a difficult point for temporal interaction prediction.

Both the RMTPP and ERPP models are developed based on the MTPP, but they have different model performances. The neural network structure and the hyper-parameters influence the final prediction accuracy.

Besides, from the loss curves in the iteration process of the RMTPP and ERPP models (see Fig. 14 and Fig. 15), it can be observed that in the first few epochs, the model loss on the validation dataset is slightly less



(a) Training loss for Scheme I:10-64-16-0.001

(b) Validation for Scheme I:10-64-16-0.001

(c) Training loss for Scheme II:10-128-16-0.001

(d) Validation for Scheme II:10-128-16-0.001

(e) Training loss for Scheme III:20-64-16-0.001

(f) Validation for Scheme III:20-64-16-0.001

Fig. 14. The iteration results of the RMTPP model under different hyper-parameter schemes *(Figure continued on next page)*.

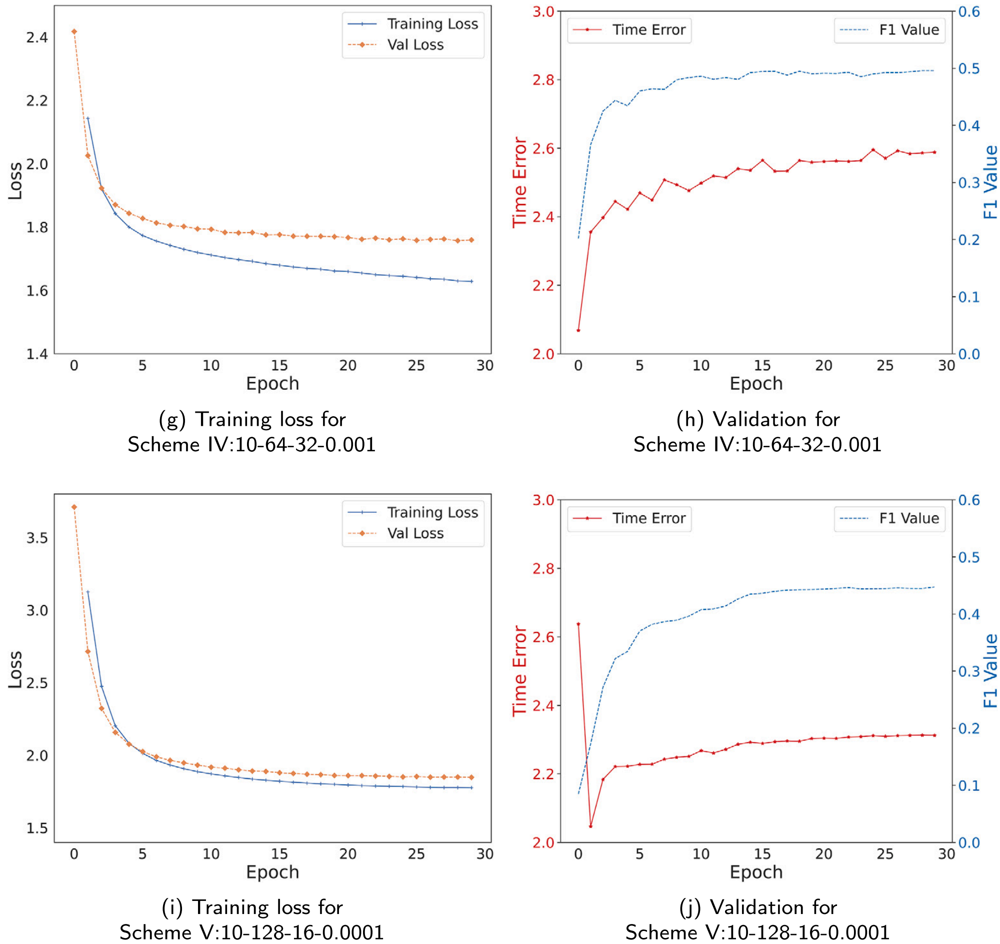
than the training loss. That is because when we train the models, the loss values of the model on the training dataset are set to be collected at the end of each batch (namely during each epoch), while those on the validation dataset are recorded after each epoch. Shifting the training loss curve 1/2 epoch to the left would make it closer to the real situation (Rosebrock, 2019), but this doesn't affect model performance and prediction results discussed above.

6. Discussion

To simultaneously predict users' next interaction and the occurrence time to that interaction, the LSTM network, RMTPP model and ERPP model are developed respectively, and their performances are compared. Through comprehensively exploring the MOOC interaction dataset and analysing the experimental results, the two research questions mentioned in Section 1 can be answered, and some insights are obtained here.

Q1: Exploration of learning patterns with temporal information

The descriptive statistics of MOOC users' usage behaviour in one month are obtained via data exploratory analysis. Combining with temporal information, the users' overall active frequency or the learning frequency for each action can be derived, which undoubtedly brings us more valuable information to understand users' learning behaviour.



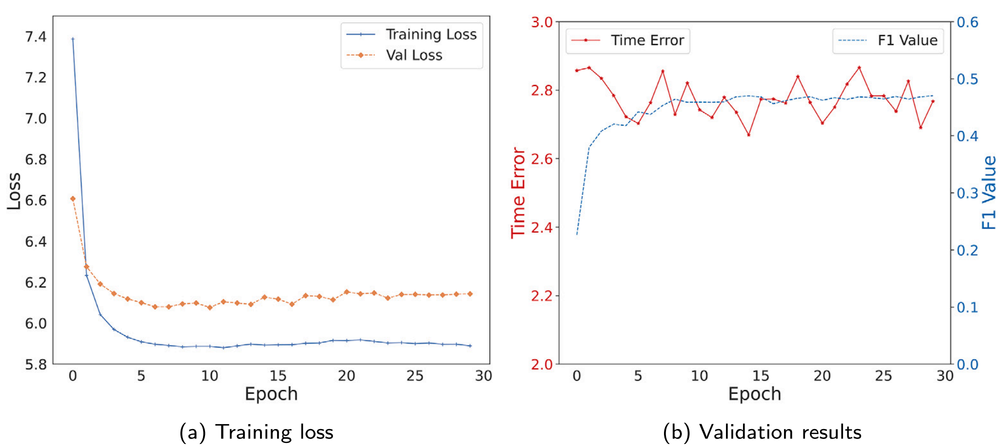
(g) Training loss for Scheme IV:10-64-32-0.001

(h) Validation for Scheme IV:10-64-32-0.001

(i) Training loss for Scheme V:10-128-16-0.0001

(j) Validation for Scheme V:10-128-16-0.0001

Fig. 14. (*continued*)



(a) Training loss

(b) Validation results

Fig. 15. The ERPP iteration results with the sequence length 10, hid_dim 64, batch size 16 and learning rate 0.001.

Besides, the distribution of the inter-action time (i.e., time interval between two adjacent actions) is another issue worthy of in-depth discussion. From the sequence plot (see Fig. 1), we can see that the time intervals between two actions are distributed unevenly. For each user, some actions are conducted in a very short period (just several seconds); however, there are also some actions that occur at long intervals. This phenomenon, which widely exists in the MOOC dataset, reflects learners' learning patterns to some extent, and it also poses a big challenge to time point prediction. On the other hand, the length of the time interval of an action might have a direct correlation with the corresponding conducted action. For example, after an action of completing a course, the time interval for the user to be active again could be relatively longer.

Q2: The correlation between action and occurrence time, and temporal model construction

As mentioned above, the learning action and its occurrence time relate closely with each other. Considering the correlation between the time and action sequences and inputting them into the neural networks simultaneously, the RMTPP and ERPP models achieve a great improvement in terms of time prediction compared with the commonly-used LSTM model. The performances of action prediction for the three models are relatively similar but the RMTPP model performs best. Therefore, we can conclude that the correlation between the action and the occurrence time point plays an important part in interaction prediction, for it helps significantly improve the accuracy of prediction.

With the appropriate hyper-parameters, the RMTPP model outperforms the two other algorithms. Applying the well-trained RMTPP model to the test dataset, the MAE of time prediction is around 2.4 (Unit: 10^4 seconds) and the action recognition rate reaches about 50% for 97 different actions. The models introduced here can realize effective MOOC temporal interaction prediction, and the marked temporal point process (MTPP) model is applicable to characterizing the temporal interaction data.

Real-world implications and instructions

In practice, accurately predicting the user's next action and its occurrence time point can help teachers in their teaching activities. For example, during the learning process of an online course, an abnormally predicted long time interval implies that students are at risk of dropping out of that course or may have relatively poor performance in the final assessments. Anomalous learning behaviours need noticing. Besides, the results of action prediction can be consulted for sequencing interactions of an online course, and then the MOOC system can generate corresponding real-time recommendations in light of the predicted user's interaction(s) at the next step or the next several steps. Combining the action prediction with the occurrence time prediction, some essential details, such as average time spent per interaction, are obtained, and then users' learning habits or learning styles can be accordingly defined.

More importantly, in terms of the recommendation system, most existing work mainly considers the next item(s) that can be recommended to users. However, the recommended time is ignored. In fact, recommending relevant items without paying attention to time points when they are really needed is not able to achieve good results. Any action needs some time to be conducted and any item needs a period of time to be used. For instance, if a user is learning a basic machine learning module on the MOOC platform, he/she would not quickly move to a more advanced module until the basic one is finished. Similarly, if a customer has bought a computer online, as time goes on, ideally, computer decorative accessories, maintenance accessories and then even a new computer product are supposed to be recommended chronologically when the customer needs them the most. Therefore, when making predictions and recommendations, it is important to consider not only the sequential nature of the data but also their temporal characteristics. Recommending an appropriate item at an appropriate time is a core and essential issue.

This paper highlights the importance of temporal information and the importance of considering the correlation between several interested indices in the prediction task. Thus, in the real-world application or future research work, multiple factors and their interactions should be focused on simultaneously and more multivariate prediction models are waiting to be proposed.

7. Conclusions

This paper focused on MOOC temporal interaction prediction, with the aim to predict a user's next action and the occurrence time to the action. Three neural network models were used: the long short-term memory (LSTM) network, and two classical marked neural TPP models, namely, Recurrent Marked Temporal Point Process (RMTPP) and Event Recurrent Point Process (ERPP).

For each algorithm, the model structure and the configuration parameters were stated. Especially for the RMTPP model, the model performances under different hyper-parameter schemes were investigated. The prediction performances of these three algorithms were then given and compared.

Through introducing the related concepts and conducting the prediction experiments, the MOOC temporal interaction prediction were investigated in depth and relatively good prediction results have been achieved. As mentioned before, the analysis of online interactions is essential for designing online courses and sequencing key interactions. In the research fields of education and learning sciences, this paper dealt with the prediction of users' interactions from a brand-new perspective of integrating temporal information and then introduced the marked temporal point process (MTPP) to describe online learning interactions. Based on the concept of the MTPP, the RMTPP and ERPP models, which are able to concatenate information on user actions and their occurrence times, were tailored and applied in the field of online education. We found that in this prediction task, the time interval between two consecutive actions is partly decided by the corresponding category of action, while the action that will be taken is relevant to the time interval/occurrence time point as well. Namely, they are inextricably linked. It provides a reference for further study that fully considering the correlation between the interested indices and other temporal/spatial features can be informative and helpful. In terms of its contributions to artificial intelligence, we authenticated the applicability of MTPP and the effectiveness of RMTPP and ERPP models on online interaction prediction compared with traditional LSTM networks. For marked neural TPP models, the layer structure and the hyper-parameters configuration were deeply explored based on a real dataset. The application scope of algorithms mentioned in the paper has been further improved.

Nevertheless, there are still some limitations in this paper. First, the corresponding action for each action ID is masked in this dataset for the sake of protecting users' privacy, so a more detailed user behavioural analysis has not been considered in this paper. Second, the proposed algorithms are applicable to temporal interaction prediction, but their performance heavily depends on the configuration parameters and model structure. This paper only considered five different schemes for the RMTPP model, and these three models are set to be implemented under a similar model structure to facilitate model comparison. Thus, the results are not necessarily the optimal ones.

In the research field of online education, MOOC temporal interaction prediction has not been concerned sufficiently. Utilizing machine learning and deep learning algorithms for such prediction is worth studying further. In our future work, we will develop novel algorithms to improve the performance of prediction.

List of acronyms

Acronym	Definition
MOOCs	Massive Open Online Courses
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Networks
TPP	Temporal Point Process
MTPP	Marked Temporal Point Process
RMTPP	Recurrent Marked Temporal Point Process
ERPP	Event Recurrent Point Process
MAE	Mean Absolute Error
MSE	Mean Squared Error

Funding

This work was supported by the Economic and Social Research Council of UK (ES/P00072X/1: 2617249, ESRC Standard Research Studentship: 22020946).

Credit author statement

The first author: Conceptualization, Data curation, Methodology, Formal analysis, Writing - original draft;

The second author: Methodology, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Statement on open data

The dataset supporting the results reported in the article is a publicly archived dataset - Social Network: MOOC User Action Dataset, which can be found in https://snap.stanford.edu/data/act-mooc.html.

Appendix A

Pseudocode of the RMTPP and ERPP algorithms

Input: Model training hyper-parameters; Model layer dimensions
Class **MTPP_RNN**(): *# Construct the model*
def _init_(*self, config, . . .*): *# Model structure*
Embedding->Embedding Dropout->LSTM->Dense layer->Dropout->Action prediction & Time prediction
def **Loss Functions**():
Action – Cross Entropy Loss
Time – RMTPP: Joint Log-likelihood Function; ERPP: MSE Loss
def **Optimizer**(): Adam
def **Train Batch**(*Batch*):
Model Outputs = model.**forward**(*Batch*)
Time Loss, Action Loss = model.**Loss Functions**(*Model Outputs*)
Total Loss = alpha * Time Loss + Action Loss
Total Loss.backward()
Optimizer()
Return Time Loss, Action Loss, Total Loss
def **Validation Loss**(*Batch*):
Model Outputs = model.**forward**(*Batch*)
Time Loss, Action Loss = model.**Loss Functions**(*Model Outputs*)
Total Loss = alpha * Time Loss + Action Loss
Return Time Loss, Action Loss, Total Loss
def **Predict**(*Batch*):
Model Outputs = model.**forward**(*Batch*)
Return Predicted Time, Predicted Action
End
def **Evaluate**(*Dataset*):
For **each** *Batch* **in** *Dataset*:
Predicted Time, Predicted Action = **Model.Predict**(*Batch*)
End
Time Error = mean(abs(Predicted Time – Actual Time))
Precision, Recall, F1 value = Compare(Predicted Action, Actual Action)
Return Time Error, Precision, Recall, F1 value
Input: Training, Validation and Test MOOC datasets
Model = **MTPP_RNN**()
For *i* **in** *1* **To** *NumEpoches*
Model.train()
For **each** *Batch* **in** *Training dataset*:
Time loss, Action loss, Total loss = **Model.Train Batch**(*Batch*)
Record Time loss, Action loss, Total loss for this *Batch*
End
Record Time loss, Action loss, Total loss for *Training dataset* in *i-th epoch*
For **each** *Batch* **in** *Validation dataset*:
Time loss, Action loss, Total loss = **Model.Validation Loss** (*Batch*)
Record Time loss, Action loss, Total loss for this *Batch*
End
Record Time loss, Action loss, Total loss for *Validation dataset* in *i-th epoch*
Time Error, Precision, Recall, F1 value = **Evaluate**(*Validation dataset*)
End
Time Error, Precision, Recall, F1 value = **Evaluate**(*Test dataset*)

References

Aktaş, D. E., & Aktaş, M. S. (2021). Sequential rule mining on the student behavior data of an e-learning platform in the field of financial sciences: Case study. In *2021 international conference on electrical, communication, and computer engineering (ICECCE)*. Kuala Lumpur, Malaysia: IEEE (pp. 1–6).

Ben Taieb, S. (2022). Learning quantile functions for temporal point processes with recurrent neural splines. In G. Camps-Valls, F. J. R. Ruiz, & I. Valera (Eds.), *Proceedings of the 25th international conference on artificial intelligence and statistics. PMLR, proceedings of machine learning research: Vol. 151* (pp. 3219–3241). https://proceedings.mlr.press/v151/ben-taieb22a.html.

Boroujeni, M. S., & Dillenbourg, P. (2018). Discovery and temporal analysis of latent study patterns in MOOC interaction sequences. In *Proceedings of the 8th international conference on learning analytics and knowledge* (pp. 206–215).

Brodahl, K. Ø., Storøy, H.-L. E., Finset, A., & Pedersen, R. (2022). The first steps towards professional distance: A sequential analysis of students' interactions with patients expressing emotional issues in medical interviews. *Patient Education and Counseling, 105*, 1237–1243. https://doi.org/10.1016/j.pec.2021.09.039.

Daley, D. J., & Vere-Jones, D. (2008). *An introduction to the theory of point processes. volume II: General theory and structure*. Springer.

Dalipi, F., Imran, A. S., & Kastrati, Z. (2018). Mooc dropout prediction using machine learning techniques: Review and research challenges. In *2018 IEEE global engineering education conference (EDUCON)*. Santa Cruz de Tenerife, Spain: IEEE (pp. 1007–1014).

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., & Song, L. (2016). 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*. ACM (pp. 1555–1564).

Enguehard, J., Busbridge, D., Bozson, A., Woodcock, C., & Hammerla, N. (2020). Neural temporal point processes for modelling electronic health records. In E. Alsentzer, M. B. A. McDermott, F. Falck, S. K. Sarkar, S. Roy, & S. L. Hyland (Eds.), *Proceedings of the machine learning for health NeurIPS workshop. PMLR, proceedings of machine learning research: Vol. 136* (pp. 85–113). https://proceedings.mlr.press/v136/enguehard20a.html.

Fatahi, S., Shabanali-Fami, F., & Moradi, H. (2018). An empirical study of using sequential behavior pattern mining approach to predict learning styles. *Education and Information Technologies, 23*, 1427–1445. https://doi.org/10.1007/s10639-017-9667-1.

Goopio, J., & Cheung, C. (2021). The MOOC dropout phenomenon and retention strategies. *Journal of Teaching in Travel & Tourism, 21*, 177–197. https://doi.org/10.1080/15313220.2020.1809050.

Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika, 58*, 83–90. https://doi.org/10.1093/biomet/58.1.83.

Hirumi, A. (2002). A framework for analyzing, designing and sequencing planned e-learning interactions. *Quarterly Review of Distance Education, 3*, 141–160.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hone, K. S., & Said, G. R. E. (2016). Exploring the factors affecting MOOC retention: A survey study. *Computers and Education, 98*, 157–168. https://doi.org/10.1016/j.compedu.2016.03.016.

Isham, V., & Westcott, M. (1979). A self-correcting point process. *Stochastic Processes and Their Applications, 8*, 335–347. https://doi.org/10.1016/0304-4149(79)90008-5.

Islam, K. T., Shelton, C. R., Casse, J. I., & Wetzel, R. (2017). Marked point process for severity of illness assessment. In F. Doshi-Velez, J. Fackler, D. Kale, R. Ranganath, B. Wallace, & J. Wiens (Eds.), *Proceedings of machine learning research. PMLR, proceedings of the 2nd machine learning for healthcare conference: Vol. 68* (pp. 255–270). https://proceedings.mlr.press/v68/islam17a.html.

Kingman, J. F. C. (1992). *Poisson processes: Vol. 3*. Clarendon Press.

Kumar, S., Zhang, X., & Leskovec, J. (2019). Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. ACM (pp. 1269–1278).

Li, S., Du, H., Xing, W., Zheng, J., Chen, G., & Xie, C. (2020). Examining temporal dynamics of self-regulated learning behaviors in STEM learning: A network approach. *Computers and Education, 158*, Article 103987. https://doi.org/10.1016/j.compedu.2020.103987.

Lien, Y.-C. N., Wu, W.-J., & Lu, Y.-L. (2020). How well do teachers predict students' actions in solving an ill-defined problem in STEM education: A solution using sequential pattern mining. *IEEE Access, 8*, 134976–134986. https://doi.org/10.1109/access.2020.3010168.

Martínez, R. S., Aricak, O. T., & Jewell, J. (2008). Influence of reading attitude on reading achievement: A test of the temporal-interaction model. *Psychology in the Schools, 45*, 1010–1023. https://doi.org/10.1002/pits.20348.

Mei, H., & Eisner, J. M. (2017). The neural Hawkes process: A neurally self-modulating multivariate point process. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems: Vol. 30*. Curran Associates, Inc. (pp. 1–24). https://proceedings.neurips.cc/paper/2017/file/6463c88460bd63bbe256e495c63aa40b-Paper.pdf.

Moore, R. L., & Blackmon, S. J. (2022). From the learner's perspective: A systematic review of mooc learner experiences (2008–2021). *Computers and Education*, *190*, Article 104596. https://doi.org/10.1016/j.compedu.2022.104596.

Naumzik, C., & Feuerriegel, S. (2022). Detecting false rumors from retweet dynamics on social media. In *Proceedings of the ACM web conference 2022*. ACM (pp. 2798–2809).

Ogata, Y. (1998). Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, *50*, 379–402. https://doi.org/10.1023/a:1003403601725.

Omi, T., ueda, n., & Aihara, K. (2019). Fully neural network based model for general temporal point processes. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems: Vol. 32*. Curran Associates, Inc. (pp. 1–11). https://proceedings.neurips.cc/paper/2019/file/39e4973ba3321b80f37d9b55f63ed8b8-Paper.pdf.

Rajendran, R., Munshi, M., Emara, A., & Biswas, G. (2018). A temporal model of learner behaviors in oeles using process mining. In *Proceedings of ICCE* (pp. 1–10).

Rosebrock, A. (2019). Why is my validation loss lower than my training loss? https://pyimagesearch.com/2019/10/14/why-is-my-validation-loss-lower-than-my-training-loss/.

Salehi, M. (2013). An effective recommendation based on user behaviour: A hybrid of sequential pattern of user and attributes of product. *International Journal of Business Information Systems*, *14*, 480–496.

Shchur, O., Biloš, M., & Günnemann, S. (2019). Intensity-free learning of temporal point processes, https://doi.org/10.48550/ARXIV.1909.12127.

Shchur, O., Türkmen, A. C., Januschowski, T., & Günnemann, S. (2021). Neural temporal point processes: A review. https://doi.org/10.48550/ARXIV.2104.03528.

Simpson, K., Beukelman, D., & Sharpe, T. (2000). An elementary student with severe expressive communication impairment in a general education classroom: Sequential analysis of interactions. *Augmentative and Alternative Communication*, *16*, 107–121. https://doi.org/10.1080/07434610012331278944.

SNAP (2019). Social network: Mooc user action dataset. https://snap.stanford.edu/data/act-mooc.html.

Tang, S., Peterson, J. C., & Pardos, Z. A. (2016). Deep neural networks and how they apply to sequential education data. In *Proceedings of the third (2016) ACM conference on learning @ scale*. ACM (pp. 321–324).

Vista, A., Awwal, N., & Care, E. (2016). Sequential actions as markers of behavioural and cognitive processes: Extracting empirical pathways from data streams of complex tasks. *Computers & Education*, *92*, 15–36. https://doi.org/10.1016/j.compedu.2015.10.009.

Wallace, R. M. (2003). Online learning in higher education: A review of research on interactions among teachers and students. *Education, Communication & Information*, *3*, 241–280. https://doi.org/10.1080/14636310303143.

Xia, X., & Qi, W. (2022). Temporal tracking and early warning of multi semantic features of learning behavior. *Computers and Education: Artificial Intelligence*, *3*, Article 100045. https://doi.org/10.1016/j.caeai.2021.100045.

Xiao, S., Yan, J., Yang, X., Zha, H., & Chu, S. (2017). Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI conference on artificial intelligence: Vol. 31* (pp. 1–8).