Stage-LSTM: Modeling Long Time Series with Arbitrary Sampling Rates and an Application on Crowdfunding

ABSTRACT

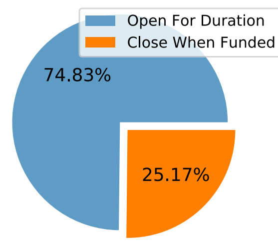
Long Short-Term Memory (LSTM) has become a standard module for time-series modeling, and it can process sequences with uniform interval quite well. However, it has at least two limitations: 1) the crucial last hidden state still cannot memorize the initial state long time ago by design; 2) it can hardly process sequences generated in different sampling rate, or cases with multiple stages within each sequence. Phased-LSTM addressed the first issue by adding a time gate to LSTM to take samples in a long sequence, while these samples still have a uniform interval. However, in Crowdfunding for example, the sequence of users' bidding behavior can be very different from the early stage to the late stage in a campaign. One should use different sampling rates for each stage of a sequence. To address these issues, we proposed an enhanced LSTM module (Stage-LSTM) equipped with a sampling gate, to model sequences from short to very long, and with arbitrary and varying sampling rates in different stages of each sequence. Experiments on two Crowdfunding datasets proved that the improvement by Stage-LSTM is significant over other methods. Based on the Stage-LSTM module, we furtherly developed a novel framework, Time-Aware Recommendation (TAR) for time-sensitive investment recommendation, by balancing Time-Aware Profits and Time-Aware Risks. Experiments demonstrated that our TAR outperforms the current state-of-the-art investment methods.

KEYWORDS

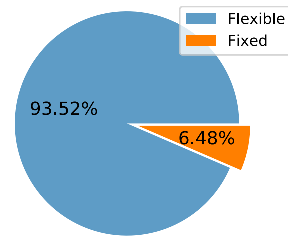
Recommender System, Deep Learning, Time Series Forecasting, Time Utility

1 INTRODUCTION

Crowdfunding is the practice of funding a project or venture by raising small amounts of money from a large number of people, typically via the Internet. The modern crowdfunding model is generally based on three types of actors: 1) creator: the project initiator who proposes the idea or campaign to be funded, 2) investor: individuals or groups who support the idea, and 3) platform: a moderating organization that brings the parties together to launch the idea. Creators apply for campaigns, and then investors bid on them for investment. Take book publication for example, authors or publishers create a campaign, and then advertise the campaign project with a subscription service. The book will be written and published once enough subscribers signal their readiness to buy the book. In this scenario, the subscribers are regarded as the investors. Thus, the list of subscribers has thus the power to create the necessary confidence among investors to finally make the campaign successful. Beside of book publication, no matter what form the return is, to get the corresponding return, the premise is that all campaigns need to be fully funded. Thus, measuring the flow risk whether a campaign can become fully-funded is important for investors. In this paper, the flow risk represents the uncertainty that a campaign cannot be fully funded. To help investors make an even better investment planning, we further build a real-time portfolio recommendation framework, which can indicate how soon will a campaign get fully funded. For a hot campaign, it is possible to get all required funds within the first day, although the designated funding time length can be one or two weeks.



(a) Distribution on Prosper



(b) Distribution on Indiegogo

Figure 1: The option distribution of campaigns.

To achieve above goals, we need to incorporates the funding dynamics. It is different from the traditional sequential prediction in which we assume uniform sampling rate - no matter how long does it take for the next event to come, one simply mark the timestamp as $t+1$. In crowdfunding dynamics, we have to take specific timestamp into account. What's more, for data collected in multiple stages, the time interval between two consecutive inputs can be uneven and vary a lot. Since a key characteristic of Recurrent Neural Network (RNN) solutions is that it mainly considers the sequential order of objects without any notion of real timestamp, a novel method which can fit the data in multiple stages is in great needed.

For debt-based crowdfunding which can return profits directly, the aim of investors is to obtain profits. The campaigns are the predecessors of loans, which are used to solicit bids from investors. If a campaign's required funds are all collected within a limited time period, it will become a loan where the investor can obtain investment returns. For the debt-based crowdfunding, we need to quantify profits and risks, which can enhance the decision-making capacities of investors.

Recently, more and more platforms support flexible funding maturity date and funding goals. If we can obtain campaigns'

XXX, 2018

ACM ISBN X-XXXXX-XX-X/XX/XX…$00.00
https://doi.org/00.0000/0000

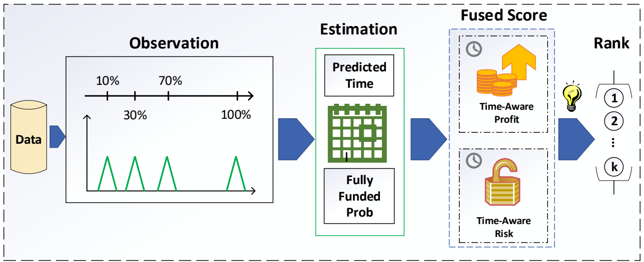


Figure 2: The flowchart of our framework.

complete timing, we are able to integrate the time utility into recommendation framework, and maximize the profits. Take Prosper for example, 25% of the campaigns shown in Fig. 1(a) have chosen the funding option "close when funded", which means that these campaigns will close as soon as they achieve their funding goals. In other words, the earlier a campaign's funding amount achieves 100%, the earlier investors can get their profits, and thus decrease their flow risks. Therefore, it is promising to exploit the time utility, which can be quantified by various statistics, such as the fully-funded time. It can be utilized to accelerate the market circulation.

In conclusion, the first goal is to build a model for all crowdfunding time series. By taking bidding's timestamp into account, the model can put similar bidding pattens within one segment automatically. Utilizing the predictive model, we then propose a quantified metric, the time utility, to take account of campaign funding time length. Furthermore, for debt-based crowdfunding, we fused the metrics of Time-Aware Profits and Time-Aware Risks within our Time-Aware Recommendation system to help investors to maximize the expected profits and minimize the risks simultaneously. The major contributions of this paper are summarized as follows.

- We proposed two enhanced LSTM modules: (1) Periodic Recurrent Neural Network (PRNN), and (2) Stage Recurrent Neural Network (SRNN). PRNN used a new linearized formulation of time gate to update the real-time oscillation period flexibly according to the timestamp of the input. While SRNN utilizes a stage detection scheme to locate the transition boundary from one periodicity to a different one.
- We devised a fused scoring model to aggregate the features of campaigns and biddings into the profits and risks framework. Then we construct a Time-Aware Recommendation (TAR) model for debt-based campaign recommendation as shown in Fig. 2 by maximizing Time-Aware Profits and minimizing Time-Aware Risks simultaneously.
- We explore the problem of quantifying the time utility from the profits and risks perspectives, and then give the top-N campaigns recommendation. Experiments were successfully carried out on two data sets in very different scales of number of campaigns.

2 RELATED WORK

In this section, we firstly introduce the characteristics of periodical crowdfunding data and highlight the importance of taking the time series into account. After that, we discuss sequential prediction and introduce the state-of-the-art Phased-LSTM (PLSTM) approach, which takes the specific timestamp to sequential prediction.

2.1 Sequential Prediction.

Sequential prediction is an ubiquitous and challenging problem that requires identifying complex dependencies between temporally correlated inputs. For instance, [22] employs Bayesian hidden Markov (BHMM) to model the campaign's market state (e.g. hot and cold). [26] used latent representations to capture spatiotemporal dynamics.

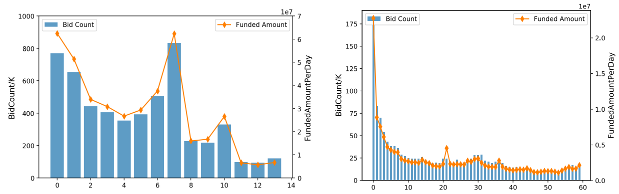
Besides, Recurrent Neural Networks (RNNs) are becoming popular building blocks for sequential prediction. For instance, [26] used RNNs for decoding states into observations. [12] and [6] use RNNs to capture the temporal dependencies by utilizing their recurrent connections.. Although RNNs have shown great superiority on sequential prediction problem, they cannot effectively learn to use the past information [1], because they usually focus on the most recent elements. In other words, one element in a sequence has more significant effect than the previous one. What's worse, RNNs cannot consider periodically recurring patterns. To solve those problems, [10] addressed the challenge of modeling long-term dependencies in sequences and [20] used deep residual network for crowd flow prediction, however, their model is not applicable for capturing patterns spanning over days or weeks as well as the temporal correlations.

Though the traditional sequential prediction is developed rapidly as discussed, it cannot apply financial behavior prediction directly, since these data base on hierarchical time series and have multi-stage property. Therefore, modeling the periodical sequence to make investment recommendation is in great need.

2.2 Background of Crowdfunding.

Crowdfunding combines the best of crowdsourcing and microfinancing, bringing together various individuals who commit money to projects and companies they want to support. It's a new and quickly growing market and it's transforming how people behave with their money. It's also transforming the ways businesses raise capital. Debt-based crowdfunding, which is also known as peer to peer lending or P2P lending, can achieve that goal. It arose with the founding of Zopa in the UK in 2005 and in the US in 2006, with the launches of Lending Club and Prosper [4]. Creators or borrowers apply online, generally for free, and their application is reviewed and verified by an automated system, which also determines the borrower's credit risk and interest rate. Investors buy securities in a fund which makes the loans to individual borrowers or bundles of borrowers. Investors make money from interest on the unsecured loans.

For the above crowdfunding product, studies focused on two important issues: predicting the funding results and identifying expected return against the risk. For the first issue, various recommender system integrate user behaviors to conduct recommendations in a target network [13, 23]. And for the second issue, multiple solutions were developed relying on different problem definitions such as a constrained portfolio selection problem [13] and an unsupervised constrained integer programming problem [5].



(a) Prosper (b) Indiegogo

Figure 3: Funding behaviors on the real-world platform. The way of segmentation to make better prediction will be addressed in Sec.6

What's more, there exist different approaches based on economics concept for recommender system. For example, [21] adapts the idea of maximizing total surplus, [18] utilizes the marginal net utility and [24] proposes multi-product utility maximization model. However, the exact funding results prediction, the utility and surplus quantification, will be tough work without the help of manual annotations. Thus, the problem of tracking accurate crowdfunding dynamics and measuring the campaign's utility automatically are under-explored.

2.3 Time Utility

Which would you rather have? $1000 today or $1000 in 5 years? If all other factors are the same, it is obvious that having $1000 today is better, which allows one to use funds for investment or consumption in advance. The time utility based on money's potential earning capacity, which means that money available at present worth more than that in the future.

However, the traditional utility-based recommendations either model users' behavior such as the rating information to predict their preferences [11, 17, 25], or ignores the sparse ratings by defining users' utility functions to better capture multiple dimensions of recommender systems [14]. Few studies take time utility into account. Since [19] declares the importance of recommending right products at right time and [15] gives rise to the demand-aware recommendation problem by combining effect of form utility and time utility. The algorithms are built for the item-based recommendation, which is different from that in the crowdfunding platform. Although [16] focuses on recommending campaigns by quantifying the time value of money in P2P online lending, the model cannot track the funding's dynamics.

Indeed, proposing a nested framework can not only track the dynamics but quantify the time utility automatically is very needed for crowdfunding. Therefore, there exists much room for improvement to achieve that goal.

2.4 Phased Recurrent Neural Network (PLSTM)

Phased-LSTM (PLSTM) [15] is a state-of-the-art RNN architecture for modeling event-based sequential data. It extends LSTM by adding the time gate k_t, which is controlled by three parameters: τ, r_{on} and s. τ represents total period of the model, s is for the phase shift and r_{on} is the ratio of the open period to the total period. All

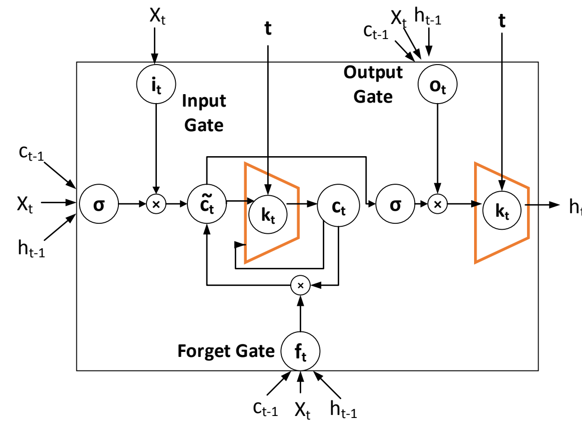


Figure 4: PLSTM's whole framework with time gate k_t, which is the same with that of our proposed model PRNN and SRNN.

of them are learned by training. k_m is formally defined as:

$$\phi_t = \frac{(t - s) \bmod \tau}{\tau}, \tag{1}$$

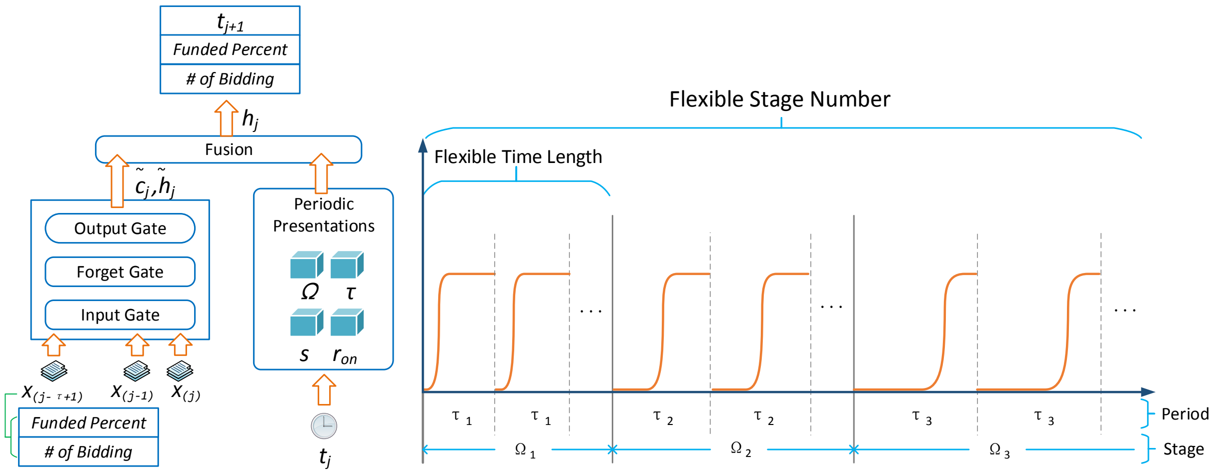
$$k_t = \begin{cases} \frac{2\phi_t}{r_{on}}, & if\ \phi_t \le \frac{1}{2} r_{on}, \\ 2 - \frac{2\phi_t}{r_{on}}, & if\ \frac{1}{2} r_{on} \le \phi_t \le r_{on}, \\ \alpha\phi_t, & otherwise \end{cases} \tag{2}$$

where t is the timestamp and ϕ_t is an auxiliary variable. The gate k_m has three phases: k_t rises from 0 to 1 in the first phase and drops from 1 to 0 in the second phase (active state). During the third phase, the model is in the inactive state. The leak rate α (close to 0 in training and equal to 0 in testing) is to propagate gradient information [7].

One key advantage of PLSTM formulation lies in the rate of memory decay. For the simple task of keeping an initial memory state c_0 as long as possible without receiving additional inputs (i.e. the input gate $i_j = 0$ at all time steps t_j), a standard LSTM with a nearly fully-opened forget gate (i.e. $f_j = 1 - \epsilon$) after n update steps would contain

$$\begin{aligned} c_n &= f_n \odot c_{n-1} = (1 - \epsilon) \odot (f_{n-1} \odot c_{n-2}) \\ &= \ldots = (1 - \epsilon)^n \odot c_0 \end{aligned} \tag{3}$$

This means the memory for $\epsilon < 1$ decays exponentially with every time step. Conversely, PLSTM state only decays during the open periods of the time gate, but maintains a perfect memory during its closed phase, i.e. $c_j = c_{j-\Delta}$ if $k_t = 0$ for $t_{j-\Delta} \le t \le t_j$. Thus, during a single oscillation period of length τ, the units only update during a duration of $r_{on} \cdot \tau$, which will result in substantially fewer than n update steps. Because of this cyclic memory, PLSTM can have much longer and adjustable memory length via the parameter τ. What's more, the oscillations impose sparse updates of the units, therefore substantially decreasing the total number of updates during network operation.



(a) SRNN (b) PRNN

Figure 5: The whole network architecture. (a) the proposed PRNN; (b) the proposed Stage-LSTM. In Stage-LSTM, the τ_1 of Ω_1 will be different from that of Ω_2. Figure (a) and figure (b) are periodic modeling for three identical sequences.

3 PROBLEM AND METHOD OVERVIEW

In this section, we first provide the problem definition in crowdfunding scenario, and then overview our proposed methods.

3.1 Problem definiton

For $I = \{1, 2, 3, ..., \|I\|\}$ campaigns with different funding duration T, we have the campaign features $X = (X_I)$ and two time series: accumulated funding amount $Y = (Y_{I,t})$ and accumulated number of bidding $B = (B_{I,t})$. Therefore, for the i-th campaign, the bidding behavior in $t_{j+\mu}$ moment can be formulated as the prediction of $(Y_{i,j+\mu}, B_{i,j+\mu})$.

The goal of PRNN and SRNN is to forecast the bidding behavior in the following period of time as well as the fully-funded time D_i of campaigns. We define the funding status prediction problem similar to next frame prediction problem in a video sequence. For simplicity of notation, we first formulate our funding status prediction problem as:

$$[X_{i,j+1}, t_{j+1}] = \sigma(W_y[X_{i,j}, c_j, h_j, t_j] + b_y), \tag{4}$$

where $X_{i,t}$ is the input features (i.e., Y_i and B_i) of the i-th campaign at a given timestamp t_j. Specifically, μ is a pre-defined sequence length which can be trained. We take the sequential features Y_i and B_i with timestamp as input. The basic block can be seen in Fig. 5 and the function works as follows:

$$\begin{pmatrix} Y_{i,j+1} \\ B_{i,j+1} \\ t_{j+1} \end{pmatrix} = \mathbb{F}(\begin{Bmatrix} (Y_{i,j-\mu+1}, B_{i,j-\mu+1}, t_{j-\mu+1}) \\ \vdots \\ (Y_{i,j}, B_{i,j}, t_j) \end{Bmatrix}), \tag{5}$$

where $Y_{i,j+1}$ and $B_{i,j+1}$ are the predictive bidding behaviors and t_{j+1} is the predicted cleander time when the next bidding happens, which is time of the predictive bidding behaviors next to t_j moment. Therefore, D_i can be observed from the predicted funding amount, i.e., once the i-th campaign's funded percent achieve 100% at timestamp t_x, we set the $D_i = t_x$.

By utilizing PRNN and SRNN result, we can quantify time utility well. Finally, we aim to determine a set of K candidate campaigns for recommendation, which is indicated by a binary decision vector y, by maximizing overall profits and minimizing total risks. Accordingly, the investment recommendation problem can be further formulated as follows, which is related to campaign's timestamp t:

$$\arg\max_{\boldsymbol{y}} \sum_{i \in I} y_i(\lambda u_i(t) - r_i(t)) \;\; s.t. \; y_i \in \{0, 1\}, \sum_{i \in I} y_i = K, \tag{6}$$

where λ is a parameter representing the weight of Time-Aware Profits $U_i(t)$ compared to Time-Aware Risks $R_i(t)$, which may vary a lot among investors.

3.2 Method Overview

Since the existing method cannot adjust the whole period of a sequence, we propose a complete model to new model to help investment decision. Fig. 2 illustrates the framework of our approach. First, we observe the data. Secondly, we proposed Periodic RNN (PRNN) and Stage RNN (SRNN), which generalized the existing model PLSTM vertically and horizontally respectively. Thus, we can predict the exact fully-funded time, which is the basis of time utility calculation. Then, we integrate time utility to the Time-Aware Recommendation (TAR) system, to get each campaign's score by Time-Aware Profits and Time-Aware Risks. Finally, we use the score to get top-N recommendation, helping investors to maximize the profits while avoiding the risks as much as possible.

4 ENHANCED LSTM MODULES

In this section, we define the structures of base prediction: Periodic RNN module (PRNN) and Stage RNN module (SRNN). The two-emphasis base forecasts have the similar parts. Thus, we detail the structures of PRNN, and then directly give the difference between PRNN and SRNN.

4.1 Periodic Recurrent Neural Network (PRNN)

PRNN proposed in this paper consists of two main components: (1) Decay control gate to capture temporal representation by taking specific timestamp into account and (2) Periodic representation parameters to dynamically fit the sequential trend from multiple periodic patterns.

4.1.1 Decay Control Gate. Capturing temporal correlations is crucial for building accurate temporal prediction models. As in many prediction problems such as funding status prediction, the funding percent at one timestamp inevitably influences nearby timestamp dynamicly. Specifically, we consider using PLSTM [15] as our recurrent neural network architecture, as it has shown good performance, but with fewer memory decay. Adding the decay control gate would result in a wider receptive field over a long period sequence and take the effective influence into account by the timestamp, which leads to better representations and more accurate predictions. Different from traditional LSTM [9], the updates to c_m and h_m are permitted only in the active state, which is controlled by k_j. The k_t is similar to PLSTM except for the definition of ϕ_j.

$$\tilde{c_j} = f_j \odot c_{j-1} + i_j \odot \sigma_c(x_j W_{xc} + h_{j-1} W_{hc} + b_c), \tag{7}$$

$$c_j = k_j \odot \tilde{c_j} + (1 - k_j) \odot c_{j-1}, \tag{8}$$

$$\tilde{h_j} = o_j \odot \sigma_h(\tilde{c_j}), \tag{9}$$

$$h_j = k_j \odot \tilde{h_j} + (1 - k_j) \odot h_{j-1}, \tag{10}$$

where i_j, f_j, o_j represent the input, forget and output gates of the j-th object respectively. Matrices W_{xc}, W_{hc} and vectors b_c are model parameters. We use σ for element-wise sigmoid function, and $\odot$ for element-wise multiplication. $\tilde{c_j}$ and $\tilde{h_j}$ are the proposed cell updates mediated by time gate k_t. The update equation of c_j has two parts, one is a fraction of the previous cell state c_{j-1} that is controlled by f_j, and the other is $\tilde{c_j}$.

From the above Prosper data analysis, we can draw a conclusion that, training a cyclic model can help improve the prediction performance of periodical data. However, To make all the segments manually, a domain knowledge about the data is required. What's worse, PLSTM can only control the global cycle of a sequence, which cannot adjust for each campaign to meet the personalized periodic modeling. Thus, we need effective periodic representations that can segment sequences the data automatically and fit different periodicity.

4.1.2 Periodic Representation. As mentioned previously, the newly launched campaign tends to attract more investors to invest, and the changes in the early period will be more intense than later. Therefore, the sampling frequency of the early stage should be relatively higher than the later stage. We should not use only one fixed parameter τ to control the whole period of the oscillation. For example, if the funding trend at dormant stage is influenced heavily by the growth stage, it could result in an inaccurate fully-funded time length prediction. Since many sequential data has the multi-stage pattern. Thus, we aim to add the periodic representation to the whole neural network mechanism in our learning procedure, which also influence the backpropagate process. The key idea in this part is to dynamically maintain memory-based parameters of periodic representations based on decay control gate, by adding two parameters a and b.

Definition 1: We define a linear periodic presentation y = $a + bn$, where a and b are two parameters while n represent the n-th period of the sequence. The way how a and b change period length is shown in Fig. **??**. Thus, we are able to update the real-time period of the oscillation flexibly according to the timestamp. For example, if t_j comes at point A in Fig. **??**, to calculate the position in the period inside, we need to subtract out the previous period firstly, which is from 1 to 3. Thus, the cutting part can be summed up by ϕ_{temp}:

$$\begin{aligned} \phi_{temp} &= s + (a + b) + (a + 2b) + (a + 3b) \\ &= s + 3a + 6b, \end{aligned}$$

When $n = 4$, we can transfer the $3a + 6b$ to $(n-1)a + \frac{1}{2}n(n-1)b$ equally. Since constants $\frac{1}{2}$ can be trained in variables b, we finally write our ϕ_m as:

$$\begin{aligned} \phi_j &= t_j - \phi_{temp} \\ &= t_j - s - (a(n-1) + bn(n-1)), \end{aligned} \tag{11}$$

ϕ_j is the auxiliary variable, which represents the period inside the rhythmic sequences. s controls the period shift of the oscillation to each LSTM cell. All parameters can be learned during the training process. When the next timestamp t_{j+1} comes, we used the p_a, p_b and s of the previous moment t_j, so that we can get the n by solving linear equations.

Though the ϕ_j is a linear function, it has powerful generalization. The a and b are trained in every cell of neural network and can be adjusted dynamically according to the prediction performance. Any sequence with increasing, decreasing or cycling trends can be well modeled, as well as the ability to quantify the trend rate of change. What's more, PRNN can learn the periodic statistical law based on ϕ_j and predict the fully-funded time effectively by synthesizing different campaigns' information. For example, for those popular, user-like campaigns, their funding will last for a short term, so their corresponding n will be smaller. Because the parameter n represents the number that a campaign will go through. Thus, we can learn when a campaign reach 100% by observing the parameter n.

4.2 Stage Recurrent Neural Network (SRNN)

There are different periodic patterns not only in different campaigns, but also in one campaign's different stages. Only use parameter a, b and s can not fit the sequence characteristics well. In order to make effective and flexible usage of periodic representations, we should merge them with current representation tensor h_m generated by the decay control gate and periodic representation unit. Thus, we propose the SRNN module not only has decay control gate, but has stage detection part.

Since we're considering multiple campaign series, which are collected by different sampling rates, automatic stage detection is a real convenience. In order to estimate relevance of each periodic representation to the current status, we use a new parameter Ω which is trained to record the stage number of campaign.

To be more specific, we have

$$\phi_j = \frac{(t - \Omega_j) \ mod \ p_j}{p_j}, \ \Omega_{j-1} \leq t \leq \Omega_j \tag{12}$$

The way how Ω works can be seen in Fig. **??**.In this method, we consider current representation tensor h_c as the context, and during training, we learn Ω for each campaign , which indicates how many stage dose a campaign contains. We require that $\Omega \leq \beta$, where β is the predefined parameter to represent the maximum number of stages. Once the stage number Ω is determined, the corresponding variable group p and s will also be determined. For example, if we predict a campaign has 4 stages, we will have a group of $\{p_1, p_2, p_3, p_4\}$ and $\{s_1, s_2, s_3, s_4\}$ simultaneously.

5 TIME-AWARE RECOMMENDATION SYSTEM

In this section, we introduce the Time-Aware Recommendation (TAR) system, which is on the basis of PRNN and SRNN. The TAR system will quantify the time utility first. And then define the Time-Aware Profits and Time-Aware Risks for the top-N recommendation.

Time-Aware Profits $U_i(t)$. In this work, we formulate Time-Aware Profits $u_i(t)$ as a combined effect of return ratio v_i (i.e. the campaign's perk or the crowdfunding interests that can get from the data directly) and time utility $\psi_i(t)$.

Since the time utility is mediated by the time elapsed since investor transfer their capital to the crowdfunding project. Let ρ_t be the daily basic rate at timestamp t. Intuitively, the more $d_i - D_{it}$, the more in advance, so will the more time utility we will get. The detailed formulas are listed as follows:

$$U_i(t) = v_i + \psi_i(t) = v_i + \rho_t(d_i - D_{it}) \tag{13}$$

$$\rho_t = \frac{\sum_{l_i}^{l_i+d_i} F(t)}{d_i}, \tag{14}$$

where $F(t)$ is the piecewise mean function for the daily base rate of Federal. l_i is the timestamp that $i-th$ campaign start for funding process, d_i is the campaign i's designated number of days to acquire the fund and D_{it} is the predicted funding usage time at t.

Time-Aware Risks $R_i(t)$. In debt-based crowdfunding, we mainly indicates two phases risks. One is the probability that a campaigns fail to be fully funded, which will lead to meaningless waiting. The other is that the investor cannot receive their return as it manifested before.Thus the Time-Aware Risks $R_i(t)$ can be translated to the joint probability p_i of the two phase risk.

$$p_i(t) = (1 - P_i(t))(1 - Q_i(t)), \tag{15}$$

where $P_i(t)$ and $Q_i(t)$ indicates the probability that campaign i fails to be fully-funded and fails to get the return at timestamp t respectively. Eq.15 tells the probability of campaign i that is not only capable to get fully funded but also capable to meet the required timeline.

Both $P(F_i(t))$ and $Q(F_i(t))$ are learned from the historical data by machine learning method *XGBoost* [2, 3]. The features used for model training are shown in Table.1. Specifically, the dynamic features obtained by the SRNN can be utilized as input features in both probabilistic prediction model, which can significantly improve model prediction performance.

Table 1: Feature examples.

Name	Description	Class
Goal	Amount the campaign aim to fund	Static
Return	Amount investor will obtain	
Category	Purpose of campaign	
City	City of creator	
Duration	Declared campaign's funding days	
...	...	
Funded_Per	Percent the campaign has funded	Dynamic
Full_Time	The predicted fully funded time	
Bid_Num	Number of the campaign's bidding	
...	...	

And the Time-Aware Risks $R_i(t)$ can be obtained as follow:

$$R_i(t) = U_i(t) \times (1 - p_i(t)). \tag{16}$$

6 EXPERIMENTS

In this section, we conduct a comprehensive set of experiments that aim to answer four key questions: (1) How does performance of SRNN and PRNN vary with the other model configurations? (2) Does our proposed Time-Aware Recommendation (TAR) system has better effect on forecasting compared with the state-of-the-art approaches? (3) Does the time aware mechanism works on the recommendation setting?

6.1 Experimental Setup

Datasets. Table 2 shows the statistics of our real-world data sets for experiments, which are collected from Prosper[1] and Indiegogo[2]. Both of them are the famous crowdfunding platform in the world. The scale of two data sets are different, while their number of time slots are in the same level. Since we aim to solve the time series forecasting problem, we can ignore the difference of campaign's and bidding number but focus on time slot itself. In order to avoid using future results, we sort the data by their timestamp first, and select the latest 20% of the data as test set. The remaining data is used for training.

Table 2: Statistics of the Dataset

Dataset	# of Campaign	# of Bidding	Time slot (h)
Prosper	117066	9336662	54840
Indiegogo	7544	1832314	41951

6.2 Evaluation Metrics

We evaluate our proposed model with two aspect protocols:

- Predicitive protocol. To show the effectiveness of our prediction method, we use Precision and Root Mean Square Deviation (RMSE) as evaluation metrics.

[1]http://www.prosper.com/.
[2]https://www.indiegogo.com/.

- **Precision.** Precision is typically used in a binary rating system. To prepare the further fully funded time prediction performance, we collected the i-th campaign's original fully funded time as s_i at first, which can be obtained directly from transaction, and then compare the corresponding predictive funding time $\hat{s_i}$ with the ground truth s_i. Finally, if $|\hat{s_i} - s_i| < \xi$, we labeled the prediction with rating 1, and rating 0 otherwise. The parameter ξ controls the fault tolerance range which can be adjusted. In this paper, it is set as 24 hours. The precision formulas is defined as:

$$Precision = \frac{E_P}{N}, \tag{17}$$

where N represents the number of the tested campaigns. E_P denotes the test campaign set whose label is 1.

- **Root Mean Square Deviation (RMSE).** To evaluate the forecasting performances, we select one of the widely-used metrics, the Root Mean Square Deviation (RMSE) for evaluation, which can reflect the actual situation of the predicted value error. Specifically, the definition is:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(s_i - \hat{s_i})^2}, \tag{18}$$

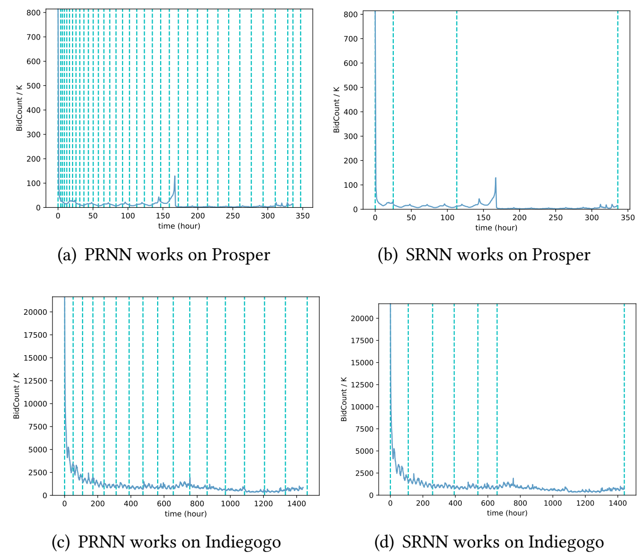
- Recommended protocol. To evaluate the performance of our proposed recommendation method, we mainly use the protocols from the economic aspect. Since in the investment recommendation, we don't tend to recommend "accurate" campaign only, but tend to recommend "right" campaign. For example, a "right" campaign means this campaign can be fully funded and will be paid off in time. To measure whether a campaign is "right", we choose the Hit-Rate (HR) and Final-Return (FR), which are for the measurement of risk and profit respectively. Generally, it is likely that the higher the HR and FR are, the more profits an investor can obtain finally.

- **Hit Rate (HR).** We adopt the percents of campaigns with positive labels on the fully-funded objectives in the selected portfolio as our evaluation metrics. Besides, we simulate the bidding processes of selected portfolios. Through simulation, we can compute the average flow risk rates of portfolios for investors theoretically, which can be treated as an overall metric considering the funding's flow risk. The formula is given below:

$$HR = Avg(\frac{|E_F|}{K} \cdot 100\%), \tag{19}$$

where E_F denotes the test campaign set which is fully-funded finally and K represents the top-K recommendation. Because the recommendations are going on and on, we select the average value as the recommended performance criteria.

- **Final-Return (FR).** No matter how we recommend campaign for investor, in debt-base crowdfunding, the return that user finally obtain should be taken into account. To calculate the Final-Return, we traced the return payments by checking the transactions. Only the campaigns whose goals are achieved and the creator fulfill their promises can the investor obtain corresponding return. And for the top-K recommendation, the metric is normalized it by FR / K. The greater the result is, the better the performance of the recommended approach.



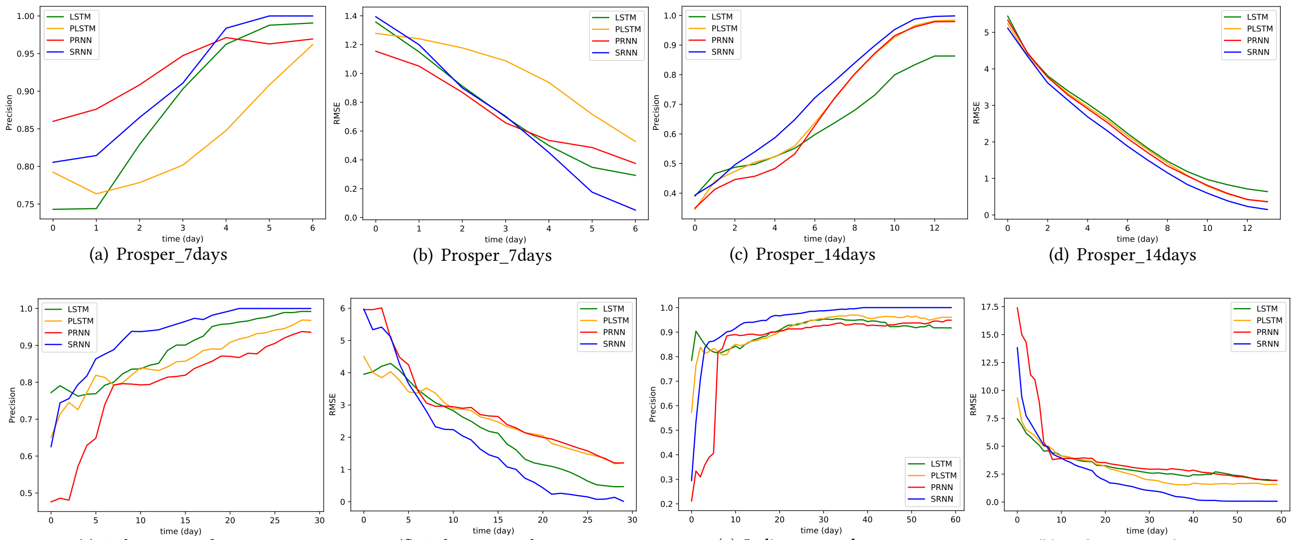
(a) PRNN works on Prosper
(b) SRNN works on Prosper
(c) PRNN works on Indiegogo
(d) SRNN works on Indiegogo

Figure 6: Sequence stage segment applications.

6.3 Baseline Algorithms

For prediction performance, we compare our model PRNN and SRNN with the LSTM and PLSTM respectively on both the precision and RMSE. And for recommendation performance, we have the TAR system, which contains the PRNN*, SRNN* and PRNN*+SRNN*. All of the three methods are on the basis of Time-Aware Profit and Time-Aware Risk. The PRNN*+SRNN* method is the linear combination of PRNN and SRNN, aiming to take advantage of both approaches. The proposed TAR system will compare with the following combination models:

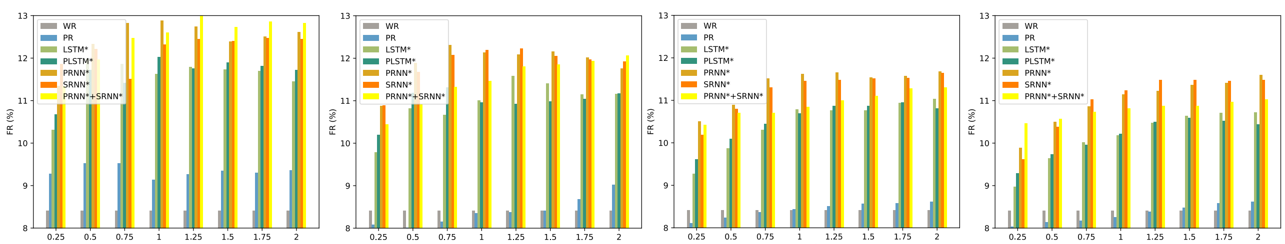
- Profit and Risk Model (PR): taking Profit and Risk into account. The traditional Profit and Risk didn't track the dynamics of funding, in that case, only static features can be used, which cannot cover time utility. Thus, the profit in PR directly refers to each campaign's declared return and the risk relative to campaign's default probability.
- LSTM based on Time-Aware Profits and Time-Aware Risks (L-PR*): training LSTM for campaign's dynamics, and then calculate fused score based on Time-Aware Profits and Time-Aware Risks, where takes time utility into account.
- PLSTM based on Time-Aware Profits and Time-Aware Risks (P-PR*): training PLSTM for the campaign's dynamics, where the other parts of model are the same as those in L-PR*.
- Without Recommendation (WR): using the actual investor's investment behaviors as comparison, which can be obtained directly by historical transaction. The model WR represents the platform's self way to recommend.



(a) Prosper_7days (b) Prosper_7days (c) Prosper_14days (d) Prosper_14days

(e) Indiegogo_30days (f) Indiegogo_30days (g) Indiegogo_60days (h) Indiegogo_60days

Figure 7: Prediction Performances on predicting fully funded time.



(a) *FR*@10 (b) *FR*@15 (c) *FR*@20 (d) *FR*@25

Figure 8: Recommendation Performances (*FR*@*N*).

6.4 Overall Performance

Prediction Performance. To validate the effectiveness of PRNN and SRNN on the precision and RMSE metric, we firstly reconstruct the campaign's fully-funded time prediction task at different timestamps. The Fig. 7 shows the forecasting performances with respect to the predicting days in both Prosper and Indiegogo data sets. Our model can automatically identify the differences between different duration campaigns, so in the training process, different duration campaigns are put together for training. However, during the test, in order to avoid the occurrence of jump layer in the images caused by the differences between the campaigns of different duration, we separated the 7 days and 14 days campaigns in Prosper as well as 30 days and 60 days campaigns in Indiegogo.

Overall, all four methods showed the same trend: RMSE decreased (in Fig. 7(b), Fig. 7(d), Fig. 7(f) and Fig. 7(h)) and Precision increased (in Fig. 7(a), Fig. 7(c), Fig. 7(e) and Fig. 7(g)) over time. The reason is that, as time goes on, the information of bidding behavior increases monotonously and the prediction uncertainty decreases as well. The more information we get, the accurate the predictions will be. From the comparisons of PSLTM and LSTM in Fig. 7(a) and Fig. 7(b), we can see the effectiveness of PLSTM only in the first days. However, for PLSTM updates the memory cell only during a small percent of the cycle and the cycle will be fixed after the train process, PLSTM's corresponding RMSE and precision have been affect obviously after the first day, where the performances are even worse than traditional LSTM. The result also proves strongly the importance of modeling dynamic period for crowdfunding series instead of fixed one. In Prosper dataset, it's PRNN that has the best prediction performance at the first forth day. For it has its own periodic representation which is shown in Fig.6(a). As in Prosper, crowdfunding lasts for a short time. In the initial stage of crowdfunding, the data are relatively dense. Thus, the PRNN outperforms SRNN whose stage granularities are much smaller. What's worse, the discarded data will reduce the accuracy of the prediction. Thus, in Fig.7(c), the prediction effect of SRNN will be worse than the LSTM that puts all the information in. However, for all the campaigns at different funding time may have different dynamic patterns, fixed period (PLSTM) and even the periodic representation fitting with linear functions (PRNN), cannot avoid the influences among different patterns but balance them by average simply. That's why the proposed SRNN for the flexible period inside the rhythmic sequences can perform stably throughout the duration.

Recommendation Performance. For recommendation performance, we compare our proposed system TAR with the baseline methods with respect to the Hit Rate (HR) and Final Return (FR) metric. The experiment is on Prosper for it's a debt-based crowdfunding data set.

All the test dataset are given random timestamps to simulate the investors bidding behaviors. At each timestamp t, the dynamic methods such as LSTM, PSLTM and our TAR system can give candidate campaigns by predicting fully-funded time, while the methods PR and WR cannot. Based on the candidate list, we finally give a top-N recommendation campaigns by utilizing the recommendation framework, i.e., L-PR*. Actually, the user preference can be simply changed by the parameter λ when recommending campaigns. Herein, we just take the average from historical records for performance comparison. In the experiment, we employ λ as 0.5, 0.75, 1, 1.25 and 1.5 respectively. And the parameter N is set to 10, 15, 20 and 25 respectively.

The results show that, the proposed TAR framework on the basis of combination of PRNN and SRNN can really enhance investment decisions, especially when the λ is set to 1.25 and the recommendation size K is set to 10. Since WR represents the return result from the platform, we can easily find that, all the other methods are outperform the WR, which show the importance of balancing the profits and risks. Except the WR, the comparison between model PR and others shows how different prediction approaches work under the Time-Aware framework, where all the latter methods take time utility into account. Since the more erroneous judgements for prediction, the more campaigns with high flow risk will put into candidates. Thus, TAR, which performs best for predicting fully-funded time, will track the dynamics closest and get good performance on FR as well. What's more, we can see the HR improvement by Table. 3. For the metric FR are prepare for profits aspect, the HR are designed for the risk aspect. No matter under which λ and K setting, our proposed TAR can get the highest HR compared with the other recommended method, which can reflects its outstanding performance in avoiding the flow risk.

Table 3: Recommendation Performance
The performance on Hit Rate (*HR@K*), the larger the better.

Methods	λ = 0.5	λ = 0.75	λ = 1.0	λ = 1.25	λ = 1.5
parameter	K = 10				
PR	63.16	62.98	62.94	62.90	62.48
L-PR*	69.08	68.55	67.75	67.21	67.10
P-LR*	70.41	70.26	69.31	68.51	67.91
PRNN*	70.84	70.88	70.15	69.77	69.58
SRNN*	70.69	70.23	69.96	69.81	69.92
PRNN*+SRNN*	**70.95**	**70.99**	**71.11**	**71.45**	**74.34**
parameter	K = 15				
PR	61.20	61.04	60.87	60.89	60.41
L-PR*	67.86	67.65	67.48	67.48	66.84
P-LR*	69.01	68.42	68.32	67.91	67.66
PRNN*	**70.0**	69.87	69.41	69.19	69.19
SRNN*	69.52	69.59	69.77	69.59	69.24
PRNN*+SRNN*	69.9	**69.92**	**69.8**	**70.05**	**69.67**
parameter	K = 20				
PR	60.20	59.89	59.54	59.39	59.26
L-PR*	67.42	67.36	66.79	66.47	66.35
P-LR*	68.21	67.65	67.25	67.06	66.91
PRNN*	68.68	68.38	68.42	68.09	67.94
SRNN*	68.57	68.45	68.34	68.09	68.09
PRNN*+SRNN*	**69.45**	**69.37**	**69.41**	**69.56**	**69.68**
parameter	K = 25				
PR	58.69	58.70	58.47	58.35	58.33
L-PR*	66.69	66.56	65.95	65.61	65.20
P-LR*	66.87	66.85	66.34	66.17	66.02
PRNN*	67.53	67.34	66.93	66.76	66.72
SRNN*	67.88	67.57	67.13	66.99	66.98
PRNN*+SRNN*	**69.07**	**69.18**	**69.25**	**69.28**	**69.28**

7 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a comprehensive study of investment recommendation in terms of multiple criteria, in the context of crowdfunding scenario. To construct this study, we firstly analyzed the real-world financial data, which contains detailed transaction information and shows obvious multi-stage attribution in predicting the fully funded time. Based on the analysis, we proposed Periodic Recurrent Neural Network (PRNN) and Stage Recurrent Neural Network (SRNN), to segment flexible period inside the rhythmic bidding behaviors sequences automatically and solve the fully-funded time prediction problem. Then, we proposed a Time-Aware Recommendation framework for debt-based crowdfunding, which can help investment decision makings. The TA framework performs a top-N recommendation by maximizing the Time-Aware Profits against the Time-Aware Risks. Experimental results have demonstrated that our proposed approaches can outperform the state-of-the-art methods.

REFERENCES

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[2] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[3] T. Chen, T. He, M. Benesty, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, pages 1–4, 2015.

[4] D. M. Freedman and M. R. Nutting. A brief history of crowdfunding. *Retrieved August*, 15:2017, 2015.

[5] Y. Fu, G. Liu, M. Teng, and C. Aggarwal. Unsupervised p2p rental recommendations via integer programming. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 165–173. ACM, 2017.

[6] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. 1999.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[8] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber. A clockwork rnn. *arXiv preprint arXiv:1402.3511*, 2014.

[11] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, and E. Chen. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1734–1743. ACM, 2018.

[12] Z. C. Lipton, J. Berkowitz, and C. Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[13] Q. Liu, G. Wang, H. Zhao, C. Liu, T. Xu, and E. Chen. Enhancing campaign design in crowdfunding: A product supply optimization perspective. pages 695–702, 2017.

[14] O. Morgenstern and J. Von Neumann. *Theory of games and economic behavior*. Princeton university press, 1953.

[15] D. Neil, M. Pfeiffer, and S.-C. Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*, pages 3882–3890, 2016.

[16] Y. Tan, X. Zheng, M. Zhu, C. Wang, Z. Zhu, and L. Yu. Investment recommendation with total capital value maximization in online p2p lending. In *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)*, pages 159–165. IEEE, 2017.

[17] M. Wan, D. Wang, M. Goldman, M. Taddy, J. Rao, J. Liu, D. Lymberopoulos, and J. McAuley. Modeling consumer preferences and price sensitivities from large-scale grocery shopping transaction logs. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1103–1112. International World Wide Web Conferences Steering Committee, 2017.

[18] J. Wang and Y. Zhang. Utilizing marginal net utility for recommendation in e-commerce. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1003–1012. ACM, 2011.

[19] J. Wang and Y. Zhang. Opportunity model for e-commerce recommendation: right product; right time. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 303–312. ACM, 2013.

[20] J. Zhang, Y. Zheng, and D. Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, pages 1655–1661, 2017.

[21] Y. Zhang, Q. Zhao, Y. Zhang, D. Friedman, M. Zhang, Y. Liu, and S. Ma. Economic recommendation with surplus maximization. In *Proceedings of the 25th International Conference on World Wide Web*, pages 73–83. International World Wide Web Conferences Steering Committee, 2016.

[22] H. Zhao, Q. Liu, H. Zhu, Y. Ge, E. Chen, Y. Zhu, and J. Du. A sequential approach to market state modeling and analysis in online p2p lending. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.

[23] H. Zhao, H. Zhang, Y. Ge, Q. Liu, E. Chen, H. Li, and L. Wu. Tracking the dynamics in crowdfunding. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 625–634. ACM, 2017.

[24] Q. Zhao, Y. Zhang, Y. Zhang, and D. Friedman. Multi-product utility maximization for economic recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 435–443. ACM, 2017.

[25] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, and D. Cai. What to do next: Modeling user behaviors by time-lstm. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3602–3608, 2017.

[26] A. Ziat, E. Delasalles, L. Denoyer, and P. Gallinari. Spatio-temporal neural networks for space-time series forecasting and relations discovery. In *Data Mining (ICDM), 2017 IEEE International Conference on*, pages 705–714. IEEE, 2017.