



## Article

# A Deep Learning Approach with Feature Derivation and Selection for Overdue Repayment Forecasting

Bin Liu <sup>1</sup>, Zhexi Zhang <sup>1</sup>, Junchi Yan <sup>1</sup>, Ning Zhang <sup>1,\*</sup>, Hongyuan Zha <sup>1,2</sup>, Guofu Li <sup>3</sup>,  
Yanting Li <sup>3</sup> and Quan Yu <sup>3</sup>

<sup>1</sup> Ministry of Education Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai 200240, China; binliu\_sjtu@sjtu.edu.cn (B.L.); jerseyzhang@sjtu.edu.cn (Z.Z.); yanjunchi@sjtu.edu.cn (J.Y.); zhasjtu@sjtu.edu.cn (H.Z.)

<sup>2</sup> School of Computer Science and Technology, East China Normal University, Shanghai 200062, China

<sup>3</sup> Ping An Asset Management Co., Ltd., Shanghai 200120, China; liguofu149@pingan.com.cn (G.L.); liyanting934@pingan.com.cn (Y.L.); yuquan690@pingan.com.cn (Q.Y.)

\* Correspondence: ningz@sjtu.edu.cn

Received: 16 October 2020; Accepted: 23 November 2020; Published: 27 November 2020



**Abstract:** Risk control has always been a major challenge in finance. Overdue repayment is a frequently encountered discreditable behavior in online lending. Motivated by the powerful capabilities of deep neural networks, we propose a fusion deep learning approach, namely AD-MBLSTM, based on the deep neural network (DNN), multi-layer bi-directional long short-term memory (LSTM) (BiLSTM) and the attention mechanism for overdue repayment behavior forecasting according to historical repayment records. Furthermore, we present a novel feature derivation and selection method for the procedure of data preprocessing. Visualization and interpretability improvement work is also implemented to explore the critical time points and causes of overdue repayment behavior. In addition, we present a new dataset originating from a practical application scenario in online lending. We evaluate our proposed framework on the dataset and compare the performance with various general machine learning models and neural network models. Comparison results and the ablation study demonstrate that our proposed model outperforms many effective general machine learning models by a large margin, and each indispensable sub-component takes an active role.

**Keywords:** overdue repayment forecasting; online lending; feature derivation; machine learning; deep learning; attention mechanism

## 1. Introduction

With the development of the economy and the rising level of consumption in national standards of living, the majority of people and companies have encountered capital turnover problems and have therefore attempted to obtain a loan for consumption, capital turnover, investment, etc. Therefore, the demand for financial credit services is continuously increasing. Online lending is a convenient and fast micro-loan innovation finance mode. Platforms streamline the intermediate, tedious loan procedure, thus attracting increasing numbers of clients who intend to solve their financial difficulties.

While online lending provides customers and platforms with an effective path towards a loan transaction, various fraudulent and insecurity factors emerge as well. Almost all credit businesses face problems such as long-term liabilities, loan delinquencies and overdue repayment, which pose great challenges for risk control in online lending. When overdue behavior occurs, financial institutions can make up for loss according to some collateral in traditional credit business. In contrast, manually

handling small monetary loans with high frequency is difficult in online lending. Once frauds such as overdue repayment occur, tracking accountability and recuperating loss become difficult problems.

Repayment behaviors in online lending always accumulate in time to form a repayment behavior sequence. Analyzing the historical record of repayment behaviors can allow potential repayment behavior patterns to be discerned, thus predicting the occurrence of overdue repayment. In [1,2], the authors developed machine learning models to analyze event records, but they ignored the sequential format in these records. Deep learning models, especially sequential neural networks, have achieved remarkable performance in event sequence-related tasks [3–5]. These kinds of models are well suited to handle the massive amount of online repayment behavior data. However, as only limited features can be collected in online lending, deep neural models may easily sink into an over-fitting problem. Feature derivation by manually designing new features is commonly utilized to enlarge the feature size, but relies on human expertise.

In this paper, we publish a new dataset collected from a practical application scenario in online lending and propose a novel feature derivation and selection approach. Based on the dataset, we propose an overdue repayment forecasting method based on a fusion of deep learning models. Experiments demonstrate that our method outperforms various general machine learning models and neural network models. Furthermore, visualization and interpretability improvement work in our approach show the critical time points and causes of overdue repayment behavior. The contributions of our research can be summarized as follows:

1. We present a new dataset that originates from a practical application scenario in online lending, which can be downloaded at [https://github.com/zjersey/payment\\_overdue\\_dataset](https://github.com/zjersey/payment_overdue_dataset). Over one million repayment records of 85,000 anonymous borrowers are contained, and all the sensitive information is encrypted for confidentiality.
2. An improved feature derivation and selection method is proposed that can generate extensive, fully-combined new features and select an arbitrary number of the most significant features based on a scorecard model.
3. We introduce deep learning models into the domain of risk control in online lending; specifically, overdue repayment forecasting based on historical repayment behaviors. Our proposed architecture, namely AD-BLSTM, combines a deep neural network (DNN), bidirectional long short-term memory (LSTM) [6] (BiLSTM) and the attention mechanism [7]. DNN and BiLSTM are used to learn from the static background information and dynamic event sequence, respectively, to maximize the superiority of the two networks. The attention mechanism is introduced to weight the importance of hidden layers in LSTM and integrate them to obtain a more informative representation.
4. Experimental results demonstrate that our approach outperforms various general machine learning models and neural network models. Interpretability improvement work is implemented based on the attention mechanism and derived features. We visualize differentiated attention weights to explore the key event time steps and analyze the feature importance of derived features to determine the causes of overdue repayment.

## 2. Related Work

Overdue repayment forecasting in this paper can be regarded as an event or behavior prediction problem. Previous works on this topic have aimed at applying predictive techniques to event sequences.

### 2.1. Event Prediction

The main purpose of event prediction is to predict the occurrence and condition of future events based on a sequence of past events. Prior research works have focused on probabilistic graphical model. Becker et al. [8] propose a framework of probabilistic models and additional methods such as the EM

algorithm [9] to predict the future behavior of business process instances based on historical event data. This framework is composed of several probabilistic modules, such as the model transformation module and prediction module, which play different roles. Breuker et al. [10] develop predictive modeling techniques to describe business process behavior. Most similar probabilistic frameworks, such as those in [11–14], require the design of a complex module structure and are not end-to-end models but have good robustness and interpretability.

Machine learning is utilized after probabilistic graphical models [1]. The three prediction models of machine learning, constraint satisfaction and quality-of-service (QoS) are combined and compared in [2]. Machine learning methods such as the decision tree [15], support vector machine (SVM) [16], Bayes network [17] and cluster analysis [18] are comprehensively used and compared. The procedure of the machine learning-based predictive technique mainly contains the two steps of data preprocessing and model learning, reducing the intricacies of the probabilistic graphical model and achieving better results while maintaining interpretability.

With the development of neural networks and deep learning techniques, the recurrent neural network (RNN) [19] and long short-term memory (LSTM) [6] have exhibited powerful abilities in sequence-related tasks, especially in the realm of natural language processing (NLP) [20–22]. Event data commonly exist in the form of sequences, and so many works have focused on deep learning-based event prediction [3–5,23,24]. Evermann et al. [25] propose a process prediction method based on LSTM to predict the behavior of the running of a process. A range of similar techniques have introduced an LSTM-based deep learning approach to predict the timestamp of future behavior [26], the continuation trajectory of running cases [27], the remaining service execution times [28] and the completion properties [29].

The point process [30–33] is a solid framework for dealing with multi-dimensional event data in the continuous time domain that treats each event as a point associated with a time stamp, location and other attributes. Previous works [34–39] have associated the point process with neural networks to process event data.

## 2.2. Deep Learning in Online Lending

Deep learning has the characteristics of end-to-end learning, thus eliminating the complicated manual design process. On one hand, the feature representation ability of neural networks is extremely powerful, and so the deep learning approaches mentioned in Section 2.1 can achieve better results than general machine learning methods and probabilistic models. On the other hand, the parameters of neurons in the network can hardly represent meaningful mathematical information of the input features, and so neural networks are poorly interpretable.

Recently, some works have introduced deep learning into the online lending domain. The authors in [40] transfer the learning algorithms of LSTM, the attention mechanism and word2vec [41], which are effective in the NLP domain, into online lending and propose a credit scoring method. The online operation record data of borrowers in online lending are regarded as a sentence with multiple words. Word2vec is applied to produce latent representation embedding for the behavior record. Instead, we divide the original behavior data into static attributes and dynamic sequences. Our proposed feature derivation and selection method is applied on the static features afterwards.

The authors in [42] develop deep learning models to predict the trading volume of the online market based on the trend of change in investor sentiment. TextCNN [43] is introduced to classify the sentiment of investor comments, and LSTM is utilized to analyze the trend of the trading volume. The prediction of the daily trading volume can be regarded as a time-series problem, while in our research, overdue repayment forecasting is an event-series problem.

## 3. Proposed System

We introduce multiple deep learning models to predict future overdue repayment behavior in online lending based on previous repayment behaviors. The structure of our proposed AD-BLSTM

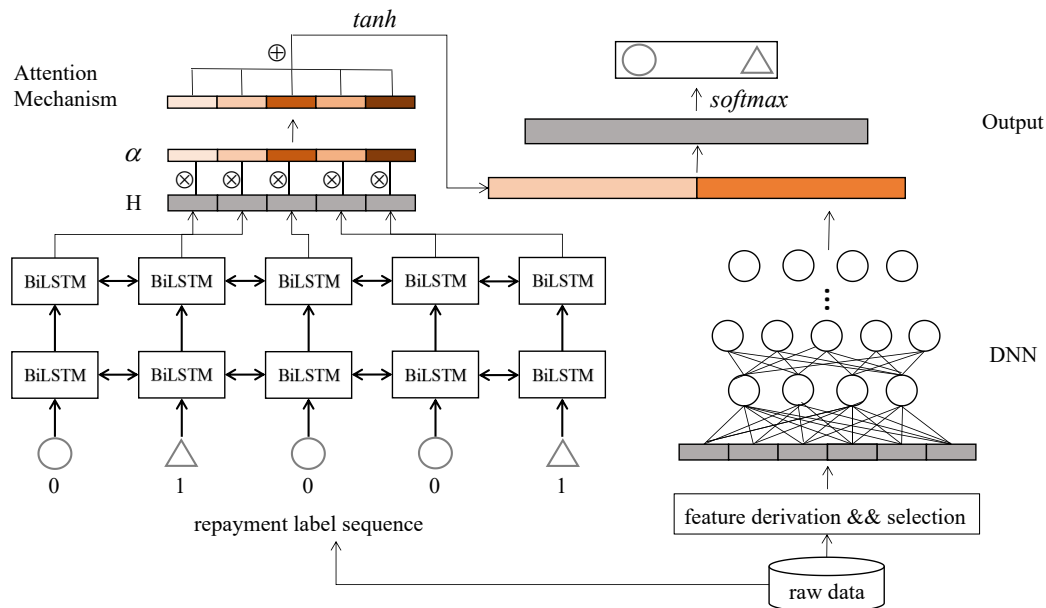
approach is illustrated in Figure 1. AD-BLSTM integrates DNN, LSTM and the attention mechanism for the purpose of appropriately representing different parts of the input repayment behavior record and improving prediction performance and interpretability. At the same time, we propose an improved feature derivation method that can generate extensive fully-combined new features and select an arbitrary number of the most significant features based on a scorecard model.

The purpose of the task is to classify future repayment behavior into two types, overdue repayment behavior (positive) and normal repayment behavior (negative), according to the past repayment logs. Therefore, the problem is simplified into a binary classification task.

As illustrated in Figure 1, the structure of our system can be divided into five modules: feature derivation and selection, a multi-BiLSTM layer, a DNN layer the attention mechanism and an output layer. We first divide the input data into dynamic and static features. The last repayment label serves as the target and all the previous time-dependent features are set as dynamic features that are fed into the multi-BiLSTM layer. Produced by feature derivation and the selection module, static features are fed into the DNN layer. We simplify the task to a binary classification task, and so the objective is to minimize the cross-entropy loss:

$$\mathcal{L} = - \sum_i [ \hat{y}^{(i)} \log(y_o^{(i)}) + (1 - \hat{y}^{(i)}) \log(1 - y_o^{(i)}) ] \quad (1)$$

where  $\hat{y}^{(i)}$  is the ground-truth and  $y_o^{(i)}$  is the output probability by the system.



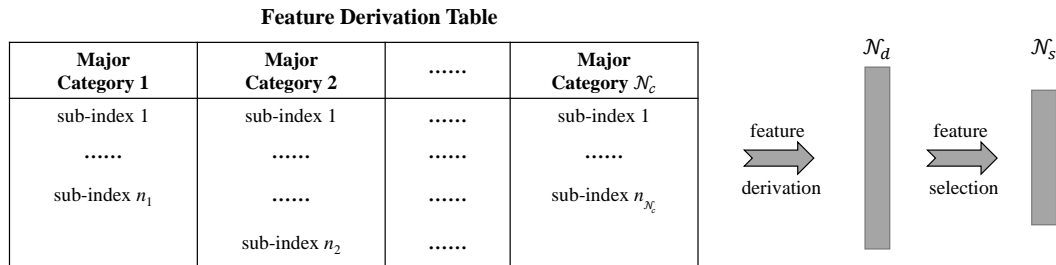
**Figure 1.** Structure of the AD-BLSTM model. Input raw data are pre-processed into dynamic label sequences and static features. Feature derivation and a selection approach are operated on static features for better informative representation. Two-layer bi-directional long short-term memory (BiLSTM) encodes the sequential input, while the deep neural network (DNN) encodes selected features. The attention mechanism balances the importance of each hidden LSTM layer, and two encoded vectors are concatenated into the output layer.

### 3.1. Feature Derivation and Selection

Generally, the process of feature derivation is essential when the features of the input are not abundant. Feature derivation requires manual design and professional prior knowledge, thus making the derivation procedure stochastic and insufficient. Generally, derivation methods are based on statistical information and expert diagnosis. Statistics-based methods calculate some common mathematical statistical values such as the maximum value, mean value and variance value of part of

the existing features. Expert diagnosis-based methods introduce new features by professional prior knowledge and manual inference.

Motivated by the statistics-based feature derivation methods and the problem of insufficiency, our approach improves upon the original method by categorizing features into various major categories and expanding the mathematical statistical values in each major class. Our proposed feature engineering framework is illustrated in Figure 2.



**Figure 2.** Overview pipeline of our proposed feature engineering method. A feature derivation table is manually designed by filling with  $\mathcal{N}_c$  major categories, each with  $n_i$  sub-indexes. The feature size is enlarged into  $\mathcal{N}_d$  based on the table. Weakly influential features are filtered out during feature selection, and the feature size is reduced into  $\mathcal{N}_s$ .

The purpose of feature derivation is to extend the number of input features from  $\mathcal{N}_0$  to  $\mathcal{N}_d$ , where  $\mathcal{N}_0$  is the number of input features and  $\mathcal{N}_d \gg \mathcal{N}_0$ . First of all, we manually design  $\mathcal{N}_c$  major categories according to the content of the input features. Besides this, mathematical statistics are set as one major category. Specifically, in this research, we set financial indicators, products, mathematical statistics, periods and time conditions as our major categories. Secondly, we add a number of subindexes as adequately as possible in each major category. For instance, the category of mathematical statistics includes subindexes of the cumulative value, cycle proportions, variance value, etc. Finally, the connection of one subindex from each major category is extracted as a new derived feature. Each customer's repayment behavior sequence is mapped into the derived features, and the mapping value is the corresponding feature value. A negative number is filled as a missing value signal when there is no accurate mapping value between the input data and derived feature.

The total amount of newly derived features can be calculated by Equation (2):

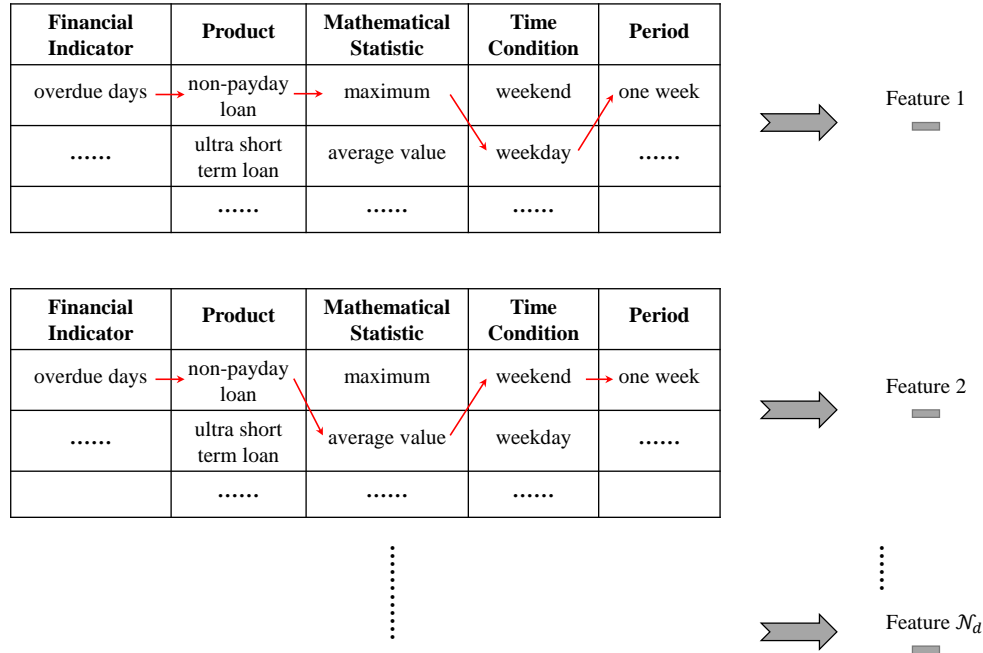
$$\mathcal{N}_d = \prod_{i=1}^{\mathcal{N}_c} n_i \quad (2)$$

where  $n_i$  is the amount of subindexes in the  $i$ -th major category. To maximize the size of derived features,  $n_i$  is supposed to be relatively large when designing the major categories and subindexes.

One specific instance is illustrated in Figure 3. By combining the “overdue days” subindex in the financial indicator major category, “non-payday loan” in product, “maximum” in mathematical statistics, “weekday” in time condition and “one week” in period, we can obtain a newly derived feature (Feature 1): the maximal overdue days for a non-payday loan on weekdays of the recent one week.

Our proposed derivation approach involves the standardization and extension of common measures, and the problem of aimlessly selecting combining variables is solved by our method. Although feature combinatorial representation can be accomplished by the neural network automatically due to deep learning's characteristic of end-to-end learning and its powerful representation ability, the experimental result in our research shows that the derived features can improve the model performance despite requiring additional work. More importantly, each new feature is produced through the connection of several indicators, which is meaningful for improving the interpretability.

After the process of feature derivation, the size of the input features is expanded from  $\mathcal{N}_0$  to  $\mathcal{N}_d$ . To eliminate the existence of meaningless padding values and weakly influential features, feature selection is essential. The feature selection approach is introduced in Algorithm 1. The selecting pipeline can be divided into four parts: chiMerge, weight of evidence (WOE), Pearson correlation coefficient and LR. After the multi-step procedure of selecting influential features, the total volume of features decreases from  $\mathcal{N}_d$  to  $\mathcal{N}_s$ . Selected features will be fed into the follow-up networks.



**Figure 3.** An example to illustrate our proposed feature derivation method. We iteratively traverse all the feature collections in the feature derivation table to enlarge the feature size. At each iteration, one sub-index is selected from each major category to form a new feature.

---

**Algorithm 1:** Feature Selection.

---

**Input:** derived features  $\lambda = \{\lambda_i\}_{i=1}^{\mathcal{N}_d}$ , thresholds  $\theta_\chi, \theta_r, \theta_p$

**Output:** selected features  $\lambda' = \{\lambda_i\}_{i=1}^{\mathcal{N}_s}$

1 chiMerge [44] ( $\lambda$ )  $\rightarrow \{\chi_i\}_{i=1}^{\mathcal{N}_d}, \{bin^{(j)}\}_{j=1}^m$

2 **for**  $i \leftarrow 1$  to  $\mathcal{N}_d$  **do**

3     **if**  $\chi_i < \theta_\chi$  **then**

4          $\lambda \leftarrow \lambda - \{\lambda_i\}$

5 WOE ( $\lambda, bin$ )  $\rightarrow \omega$

6 **for**  $\omega^{(i)}, \omega^{(j)}$  in  $\omega$  **do**

7     Pearson correlation coefficient:

$$8 \quad r_{i,j} = \frac{\sum_k (\omega_k^{(i)} - \bar{\omega}^{(i)}) (\omega_k^{(j)} - \bar{\omega}^{(j)})}{\sqrt{\sum_k (\omega_k^{(i)} - \bar{\omega}^{(i)})^2 \sum_k (\omega_k^{(j)} - \bar{\omega}^{(j)})^2}}$$

9     **if**  $|r_{i,j}| > \theta_p$  **then**

10          $\omega \leftarrow \omega - \{\omega^{(i)}, \omega^{(j)}\}$

11 Logistic Regression: LR ( $\omega$ )  $\rightarrow p$ -value, coef

12  $\zeta \subset \omega$  s.t.  $p$ -value( $\zeta$ )  $> \theta_p$

13  $\omega = \omega - \zeta$

14  $\lambda' \subset \omega$  s.t. coef( $\lambda'$ ) has the same sign

---

### 3.2. DNN Layer

The deep neural network (DNN), also known as the multi-layer perceptron neural network (MLP), is the most fundamental network. A large DNN is stacked by precursor perceptron neurons with weights and an activation function. A single perceptron cannot represent a linearly non-separable situation, even the basic logic operation “xor”. Expanding the number of perceptrons and connecting layers can represent any mathematical function.

The relationship between the input  $x^{(i)}$  and output  $y^{(i)}$  of the  $i$ -th layer can be calculated by Equation (3):

$$y^{(i)} = \mathcal{F}(\mathbf{W}^{(i)}x^{(i)} + \mathbf{b}^{(i)}) \quad (3)$$

where  $\mathcal{F}$  is the activation function, usually tanh, ReLU or the sigmoid function.  $y^{(i)}$  can be the output possibility or the input of the next layer as  $x^{(i+1)}$ .

We train a three-layer DNN as an encoder of input features  $\mathbf{X}_s \in \mathbb{R}^{\mathcal{N}_s}$  after derivation and selection into vector  $\mathcal{M} \in \mathbb{R}^p$ , where  $p$  is the number of neurons in the last layer.

### 3.3. Multi-BiLSTM Layer

In neural networks such as DNN and CNN, the inputs are independent of each other without temporal dependence, while the recurrent neural network (RNN) considers sequential information in which the data are not only related to the input at this time but also related to the previous input. In other words, the RNN has the ability to memorize. The RNN models the dependency within sequence data extensively, but the accumulation of the gradient product of each time step in backward propagation causes the gradient to disappear when the sequence length is long.

On the basis of the RNN, LSTM is a type of neural network with the additional ability of forgetting, which is thus suitable for sequence data and has achieved outstanding results in natural language processing. The shortcoming of the RNN is that there is only one hidden layer state updating inside the network, so the model is relatively simple. All the historical input data are memorized by the RNN without filtering, thus frequently resulting in the long-distance dependence problem. LSTM alleviates the gradient disappearance and explosion problem by introducing several gate units with different functions. LSTM selectively operates on input information which is memorized, forgotten or output to the next layer with a certain weight according to the content importance of the information. All the operations are implemented by multiple computing components called “cell gates”, including forget gates, input gates and output gates.

Firstly, the forget gate calculates the degree of forgetting the historical information, denoted as  $\mathbf{C}_t$ , based on the input data and the hidden state of the previous moment.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, x_t] + \mathbf{b}_f) \quad (4)$$

$\mathbf{f}_t$  will be multiplied by  $\mathbf{C}_{t-1}$  afterwards to forget partial information in  $\mathbf{C}_{t-1}$ .

The input gate updates the candidate hidden layer value and the hidden layer state. Referring to Equation (5),  $\mathbf{i}_t$  determines the proportion of candidate hidden values updating to the hidden value  $\mathbf{C}_t$ , and the candidate hidden value  $\widetilde{\mathbf{C}}_t$  is calculated by Equation (6). The hidden layer state  $\mathbf{C}_t$  is updated with the forget gate, based on Equation (7).

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, x_t] + \mathbf{b}_i) \quad (5)$$

$$\widetilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, x_t] + \mathbf{b}_C) \quad (6)$$

$$\mathbf{C}_t = \mathbf{f}_t * \mathbf{C}_{t-1} + \mathbf{i}_t * \widetilde{\mathbf{C}}_t \quad (7)$$

The function of the output gate is to output the hidden state in a certain proportion.

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, x_t] + \mathbf{b}_o) \quad (8)$$

$$\mathbf{h}_t = \mathbf{o}_t * \tanh(\mathbf{C}_t) \quad (9)$$

Historical information does not flow into the future state entirely, while essential information is preserved and useless information is forgotten after the three cell gates.

LSTM and the RNN both process sequence data in one direction, and  $\mathbf{h}_t$  is determined only by  $x_t$  and  $\mathbf{h}_{t-1}$ , ignoring the correlation between future events and current events. BiLSTM complements LSTM to address this problem of using a single direction. By contrast, the input sequence is processed in the reverse order simultaneously to obtain another hidden state,  $\mathbf{h}_{tb}$ , thus representing a sequence in two directions, and two hidden states are concatenated afterwards to get the final hidden state  $\mathbf{h}'_t = [\mathbf{h}_{tf}; \mathbf{h}_{tb}]$ .

Consider a sequence of overdue repayment flags of length  $L$ :  $\mathbf{X} = \{x^{(1)}, x^{(2)}, \dots, x^{(L)}\}$ , where each point  $x^{(t)} \in \{0, 1\}$  in the sequence represents whether the  $t$ -th repayment behavior is overdue (when  $x^{(t)} = 1$ ) or not (when  $x^{(t)} = 0$ ). Motivated by the structure of the pretrained language representation model ELMO [45], we train a two-layer BiLSTM network as the encoder to reconstruct input sequential behavior data into the vector  $\mathcal{H} \in \mathbb{R}^{2m \times L}$ :

$$\mathcal{H} = [ [\mathbf{h}_{1f}; \mathbf{h}_{1b}], [\mathbf{h}_{2f}; \mathbf{h}_{2b}], \dots, [\mathbf{h}_{Lf}; \mathbf{h}_{Lb}] ] \quad (10)$$

where  $\mathbf{h}_{if} \in \mathbb{R}^m$  and  $\mathbf{h}_{ib} \in \mathbb{R}^m$  are the forward and backward hidden states, respectively, in the  $i$ -th time step of the second LSTM layer.

### 3.4. Attention Layer

We propose an attention mechanism operating on the hidden state at each moment. We calculate the importance weight of each hidden state and combine the states based on weights to obtain a combined final hidden state. The formulas of the attention mechanism are listed below:

$$\mathbf{M}_h = \tanh(\mathcal{H}) \quad (11)$$

$$\boldsymbol{\alpha} = \text{softmax}(\boldsymbol{\omega}^T \mathbf{M}_h) \quad (12)$$

$$\mathbf{r} = \mathcal{H} \cdot \boldsymbol{\alpha}^T \quad (13)$$

$$\mathbf{h}^* = \tanh(\mathbf{r}) \quad (14)$$

where  $\mathcal{H}$  is the output of the BiLSTM layer referring to Equation (10) and  $\boldsymbol{\omega} \in \mathbb{R}^{2m}$  is the variable learned by the training process.  $\boldsymbol{\alpha} \in \mathbb{R}^L$  reflects the weights of hidden states at different moments and is operated on  $\mathcal{H}$  to obtain the final output state  $\mathbf{h}^* \in \mathbb{R}^{2m}$ .

### 3.5. Output Layer

We concatenate the output of the multi-BiLSTM layer  $\mathbf{h}^*$  and the DNN layer  $\mathcal{M}$  to form a vector  $\mathbf{X}_o \in \mathbb{R}^{2m+p}$  and apply it to a fully-connected layer and softmax function to obtain the predicted probability  $y_o$ .

## 4. Experiments

### 4.1. Dataset

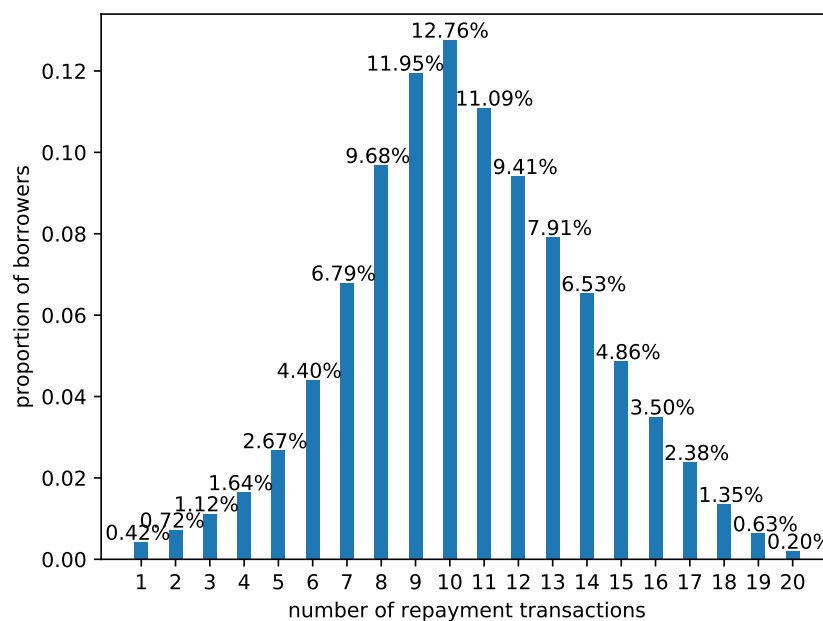
Customers normally submit their fundamental information, such as identity information, wealth information and credit records, when they borrow money from online lending platforms. Besides, online lending platforms may record transaction details when loan behaviors are continued.

We present a new dataset in this paper, which is available to the public ([https://github.com/zjersey/payment\\_overdue\\_dataset](https://github.com/zjersey/payment_overdue_dataset)). The real-world dataset was provided by a company in Shanghai and collected from the two situations above. The practical situation is that of an online lending platform that is used by a large number of borrowers and lenders. Borrowers and lenders can engage in loan transactions with each other under the control of the platform. Our dataset is sampled from the transaction records of the platform.

Each piece of data represents a record of a repayment transaction from a borrower to a lender. Our dataset contains 1,048,575 transaction records and 85,236 borrowers are involved, with an average of 12.3 repayment behaviors for each borrower. The maximal number of repayment records for a borrower is 20. The number of borrowers with a different number of records (ranging from 1 to 20) was counted, and the proportion is illustrated in Figure 4. Since the length of the behavior sequence in our dataset is not particularly long and there are borrowers with few records, some long-sequence modeling based methods [6,19] would not perform well for our dataset.

As a borrower might have transactions with multiple lenders at the same time, the repayment behavior does not have a regular frequency in our dataset. Besides, repayment behaviors between precise borrower-to-lender matches may be small in number. Therefore, our proposed approaches focus on modeling the historical records from the borrower level instead of the borrower-to-lender level.

Each transaction record contains 64 engineering features, which can be divided into three categories based on content:



**Figure 4.** Statistical illustration of the proportion of borrowers with different numbers of repayment records.

- Customer information (after data masking): borrowers' unique identification, industry, etc. Note that data masking has been applied to protect the privacy of customers.
- Dynamic repayment records: obliged repayment amount, obliged repayment time, overdue days, etc.
- Background information: expense ratio, loan type, order scenario, etc.

The most meaningful information lies in the dynamic features. Each repayment transaction record has a due date, due amount of money, actual repayment date and actual repayment amount. The transaction record is identified as an overdue repayment if the actual repayment date is later than the due date or the actual repayment amount is less than the due amount. The background and identity features provide some auxiliary information. Although there are 64 features in total, some of them are

meaningless or have a strong correlation. Therefore, the workable features after feature screening are limited in number, which makes feature engineering essential and challenging.

#### 4.2. Data Preprocessing and Experimental Settings

We define a record as overdue repayment behavior if the overdue day or overdue amount in a repayment record is not equal to zero. The label for overdue repayment is set as positive; otherwise, it is set as negative. To define the target of the training process, we first sort the data by unique identification and a repayment timestamp. The target of the individual with the last repayment behavior being overdue is set to positive; otherwise, it is set as negative. As a result, the number of positive individuals is 15,608, accounting for 18.3% of the total, while negative samples account for 81.7%.

The sequence of all the behavior labels except the last label is regarded as the dynamic features and fed into the multi-BiLSTM layer. The window length of the label sequence is the maximum of each individual's record length, and padding is added at the end of the sequence to make up the required length. We operate a feature derivation and selection approach on the records, except the last record, for each individual to obtain static features as the input of the DNN layer. In feature derivation, we design five major classes as described in Section 3.1 with 14, 7, 20, 3 and 10 subindexes, respectively; thus, the total amount of newly derived features is  $\mathcal{N}_d = 14 \times 7 \times 20 \times 3 \times 10 = 58,800$ . After feature selection, we retain 42 of the most influential features in the DNN layer.

We choose TensorFlow (<https://www.tensorflow.org/>)—a deep learning tool based on Python—as the deep learning framework for our experiment. We randomly select 80% of the data as the training set, 10% as the validation set and 10% as the test set. The parameters of our AD-MBLSTM model are listed in Table 1.

**Table 1.** Parameter setting.

Parameter	Parameter Description	Value
time_step	Length of input sequence	15
lr	Learning rate	0.03
optimizer	Optimization method	Adam
lstm_unit	Neuron number in single LSTM	50
DNN_units	Neuron numbers in hidden layers of DNN	[150, 300]
epoch	Training rounds	20
batch_size	Batch size	128
dropout	Dropout ratio of LSTM	0.15

#### 4.3. Evaluation Indicators

The statistics of the positive and negative sample proportions described in Section 4.2 indicate that the data distribution is unbalanced with few positive samples. Supposing that every input individual is classified as negative, then we can still obtain an accuracy of 81.7%. Obviously, this accuracy value is quite high but meaningless, because no overdue behavior can be recognized by the model. As the performance of the model cannot be measured appropriately only in terms of accuracy, we supply other evaluation indicators including recall, the area under the curve (AUC) value and KS value.

The confusion matrix is a fundamental evaluation indicator in binary classification tasks and is essential for the calculation of multiple other indicators as well. The confusion matrix is a  $2 \times 2$  table with four combinations of actual values and prediction values:

- TP: True positive, showing that the actual value and prediction value are both positive.
- TN: True negative, showing that the actual value and prediction value are both negative.
- FP: False positive, showing that the actual value is negative while the prediction value is positive.
- FN: False negative, showing that the actual value is positive while the prediction value is negative.

On the basis of the four combination values above, TPR, TNR, FPR and FNR are the four ratio values that represent the proportion of corresponding combination values. The formulas for TPR and FPR are as follows:

$$TPR = \frac{TP}{TP + FN} \quad (15)$$

$$FPR = \frac{FP}{TN + FP} \quad (16)$$

All of the evaluation indicators can be calculated by the combination values and ratio values mentioned above. The recall value represents the proportion of the positive samples that are predicted as the correct class, according to the following formula:

$$recall = \frac{TP}{TP + FN} \quad (17)$$

where a higher recall value indicates that more positive samples are distinguished by the model, which satisfies the actual demand of our research; i.e., to distinguish overdue repayment behaviors.

Gradually changing the threshold of classification from 0 to 1 and calculating the corresponding TPR and FPR, we can form a line graph regarding the (TPR, FPR) pairs as points, named the receiver operating characteristic (ROC) curve. The AUC is the area under ROC curve. A large AUC value indicates good performance of the classification model, and a perfect model has an AUC of 1.

Similarly, taking the threshold as the x-axis and the TPR and FPR as the y-axis, we can obtain a graph with two lines, named the KS curve. The KS value is the maximum distance between the FPR curve and FPR curve in the vertical direction.

$$KS = \max(|TPR - FPR|) \quad (18)$$

where a higher KS value indicates that positive and negative samples are distinguished more obviously.

#### 4.4. Baselines

In order to comprehensively measure the performance of our proposed AD-MBLSTM model, we use baseline models, including multiple general machine learning models and basic neural networks, that have been utilized frequently in previous work for comparison. We divide the input data into dynamic and static features in AD-MBLSTM; however, this operation does not suit the baseline models, so we simply implement feature derivation and selection on the inputs and afterwards train the processed data using baselines with the same parameters.

- Logistic regression (LR) is a fundamental and commonly used classification approach based on sigmoid function and the maximum likelihood method. The LR model does not need to assume the prior distribution of input data. Not only can the classification label be determined, but the predicted probability can also be obtained. and so the threshold can be adjusted according to demand and label distribution.
- XGBoost [46] achieved state-of-the-art performance in large numbers of machine learning tasks as soon as it was proposed. The overall idea of XGBoost is to constantly add new decision trees to improve the performance of a system. Newly supplied trees can make up for the shortcomings of the previous weak classifiers to compensate for prediction residuals.
- The factorization machine (FM) [47] interactively combines input features in pairs, which allows training to be performed on each pair of potential features. There is a similarity between FM and SVM in the formula. However, in contrast to SVM, all interactions between features are considered via factorized parameters in FM, and so FM works well even in problems with huge sparsity.
- DNN extracts derived and selected features directly and connects them to the output layer.

- In multi-BiRNN, we consider each variable of the input record as a dynamic variable and feed input data into a two-layer bidirectional RNN. We choose this structure due to its similarity to the LSTM layer in AD-BLSTM, thus facilitating comparison.
- Multi-BiLSTM has the same structure as multi-BiRNN except that the RNN is replaced with LSTM.

#### 4.5. Result Analysis

The comparison results of AD-BLSTM with baseline methods are listed in Table 2. Our proposed AD-BLSTM model can be seen to outperform other methods in all evaluation indicators, especially in recall, AUC and KS values. AD-BLSTM achieves 0.59 recall value on the basis of an accuracy of over 0.85, indicating that 59% of overdue repayment behaviors can be predicted accurately. The AUC and KS values of AD-BLSTM suggest that it is capable of distinguishing positive and negative samples and thus represents a stronger classifier compared with other methods.

Traditional methods regard repayment records as a single input instead of a sequence, processing data into a fixed vector via feature engineering and training by general machine learning methods or neural networks, which has a similar procedure to the LR, XGBoost, FM and DNN baselines. By observing these baseline results, we can conclude that even the procedure of feature engineering has been improved by our proposed feature derivation approach, and although strong classifier algorithms such as XGBoost and FM have been utilized, the recall, AUC and KS still fluctuate to an unsatisfying level.

The training pipeline for the two memory network baselines, Mul-BiRNN and Mul-BiLSTM, shows large difference with the above four baseline methods. Repayment records are regarded as a time sequence instead of a non-sequential vector. As the recurrent network can extract sequential information effectively, the two baseline methods perform much better than traditional methods. Therefore, sequential modeling is essential in the context of our research.

**Table 2.** Comparison results with baseline models. The best is in bold. AUC: area under the curve; KS: Kolmogorov-Smirnov value; FM: factorization machine; Multi-BiRNN: multi-layer bi-directional recurrent neural network; AD-BLSTM: our model.

Methods	Accuracy	Recall	AUC	KS
LR	0.850	0.18	0.675	0.286
XGBoost	0.851	0.19	0.684	0.292
FM	0.852	0.21	0.684	0.304
DNN	0.852	0.23	0.685	0.315
Multi-BiRNN	0.798	0.43	0.779	0.432
Multi-BiLSTM	0.804	0.47	0.781	0.438
AD-BLSTM	<b>0.855</b>	<b>0.59</b>	<b>0.844</b>	<b>0.481</b>

#### 4.6. Ablation Study

In this subsection, we explore the effect of the attention mechanism, feature derivation method, sequential input features and static input features; the results are listed in Table 3. In the table, “w/o attention” refers to the concatenation of the hidden vector of BiLSTM at the last moment with the DNN-extracted feature vector connected to the output layer; “w/o derivation” refers to the operation of basic feature engineering approaches such as normalization and one-hot encoding on the first record of each individual piece of raw data and directly connecting to the DNN layer, instead of feature derivation and selection. We can conclude from Table 3 that the attention mechanism and feature derivation improve the prediction performance of AD-BLSTM, but modestly. However, in the next subsection, we will show that the two components play an essential role in interpretability. Furthermore, in the table, “w/o sequence” refers to the isolation of BiLSTM and the attention layer, degenerating to the DNN baseline in Table 2, while “w/o statistics” isolates the DNN layer for the purpose of observing the effect of sequential behavior and static background information. The results

indicate that the previous repayment label sequence has an influential impact on the next repayment target even without identification and background information, while the impact of static features is far less influential. Another observed phenomenon is that the accuracy has the opposite variation trend to the other indicators, perhaps for the reason that the model pays more attention to distinguishing overdue repayments, thus misclassifying some negative instances.

**Table 3.** Ablation study results. The best is in bold.

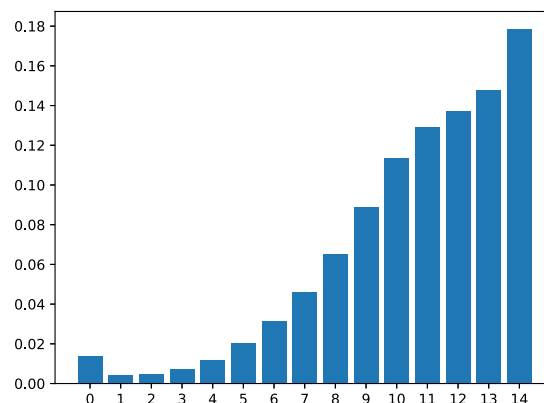
Model	Accuracy	Recall	AUC	KS
AD-BLSTM	0.855	<b>0.59</b>	<b>0.844</b>	<b>0.481</b>
↪w/o attention	<b>0.857</b>	0.57	0.843	0.478
↪w/o derivation	0.840	0.57	0.842	0.477
↪w/o attention and derivation	0.842	0.53	0.840	0.473
↪w/o sequence	0.852	0.23	0.685	0.315
↪w/o statics	0.785	0.41	0.747	0.416

#### 4.7. Interpretability

In this subsection, we introduce our work exploring the cause for overdue repayment, which can enhance the credibility of the prediction results and meanwhile provide possible prevention approaches. Our interpretability work is mainly based on the attention mechanism and feature derivation.

##### 4.7.1. Locating Critical Time Point via the Attention Mechanism

In the attention layer, the vector  $\alpha$  reflects the importance weights of LSTM hidden layers at different moments, calculated by Equation (12). Therefore, we explore the pivotal time points in a sequence of repayment actions by visualizing and analyzing the vector  $\alpha$ , as illustrated in Figure 5. We calculate the mean  $\alpha$  value of all individuals and visualize it as Figure 6. The importance weight has a positive correlation trend with the elapsing of time points except for the first time point. The reason for this is that the hidden layer representation in LSTM is calculated by the current input and previous hidden representation, thus containing more information than the previous input. However, the first hidden state is still distinctly larger than the next. We calculate the difference value of adjoining hidden states in positive samples and visualize it as Figure 5. The result demonstrates that the first and last three hidden states have more prominent weights than the others. In the process of dividing dynamic sequences, padding is performed at the end of sequences whose behavior number is less than the time\_step, and the hidden states remain unchanged in these padded moments. Therefore, the last three hidden states may all refer to the last repayment behavior. We conclude that the first and last behavior have the most significant influence on the final result.



**Figure 5.** Visualization of mean attention weights.

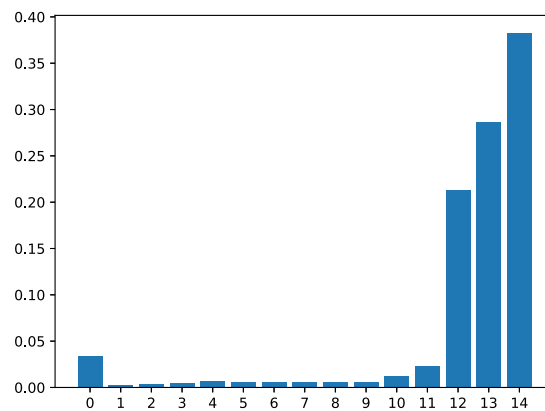


Figure 6. Visualization of differential attention weights.

#### 4.7.2. Analysis of Derived Features

The derived features can be fed into machine learning models with interpretability to further explore the latent logic of overdue repayment behavior. We choose XGBoost as the classifier model and set the features and corresponding targets as inputs. After convergence, we analyze the importance of features and explore the most meaningful subindexes.

To begin with, we select some of the static background features as the input for the XGBoost model. After convergence, the importance weights of the features are illustrated in Figure 7. In the figure, *product\_id\_0-3* stands for the four types of loan products. *sub\_industry\_name\_0-6* stands for the seven types of sub-industry and *industry\_id\_0-2* stands for the three types of industry: consumer finance, Internet finance and their integration. The meaning of all these features has been included in the description file of our dataset. From Figure 7, we can conclude that the type of product and industry has a significant impact on the overdue repayment behavior. Surprisingly, most people assume that the amount of owed money may greatly influence the overdue repayment, but as illustrated by our results, the owed money (*start\_money* in the chart) has a similarly low impact to birthday, region and gender.

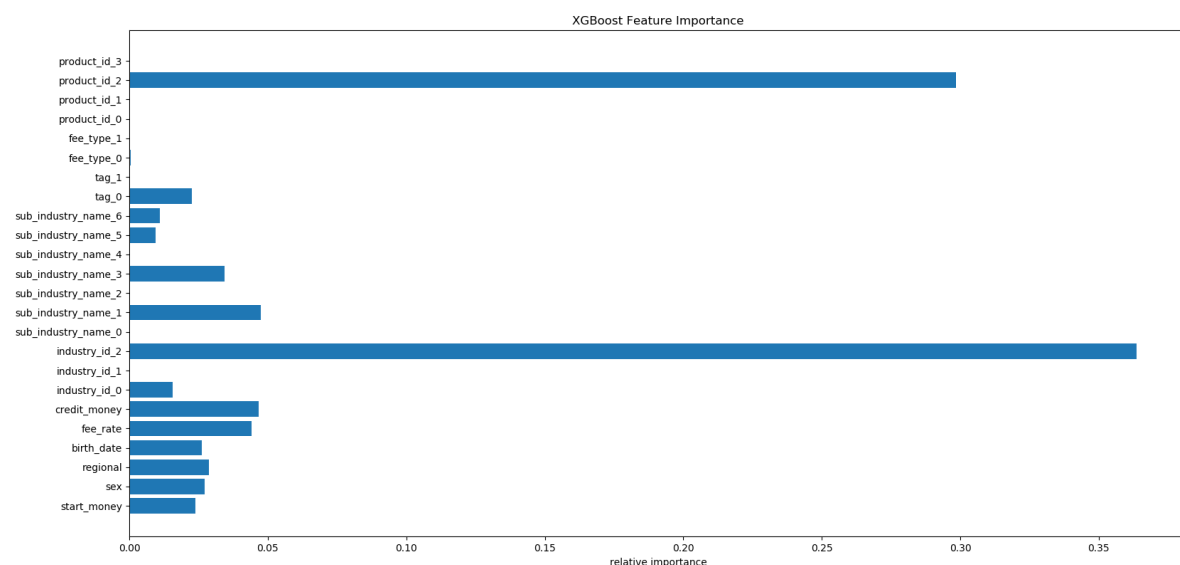
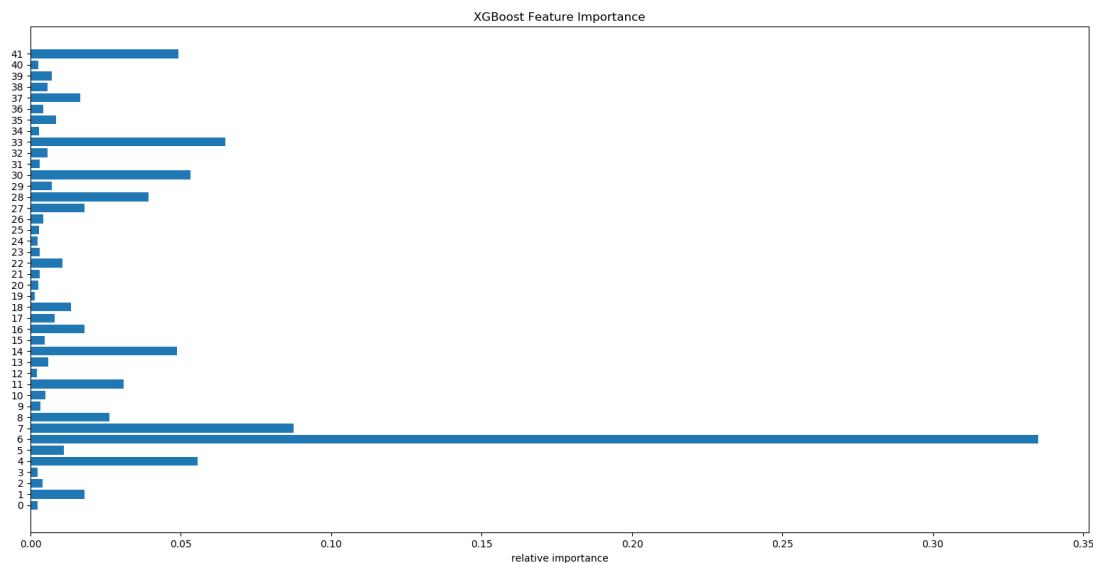


Figure 7. The importance of static background features calculated by XGBoost.

Furthermore, we analyze the importance weights of all the 42 features obtained by our proposed feature derivation and selection method. The results are illustrated in Figure 8. The longitudinal axis denotes the order number of features, and the meaning of features has been provided in our dataset. By summarizing the key words of the features with five largest importance weights, we conclude that the last behavior has more of an impact on whether a future repayment will be overdue or not

than the others, which agrees with the result after analyzing the attention weights. Additionally, whether the type of product is a very short-term cash loan matters a great deal, corresponding to the above conclusions from analyzing the background features.



**Figure 8.** The importance of 42 selected derivation features calculated by XGBoost.

## 5. Conclusions

In this paper, we proposed a fusion deep learning model and a novel feature derivation and selection approach for overdue repayment forecasting. Our methods were evaluated on a real-world dataset that we collected and made publicly available. Multiple neural networks were combined to simultaneously encode the static background information and dynamic sequential information. Experimental results demonstrated that our model outperforms various machine learning models and neural networks. The proposed feature derivation method can generate a large number of combination features from the original low-quality features. Furthermore, we visualized attention weights and found that the first and last behaviors are critical time points in a repayment behavior sequence. By analyzing the derived features, multiple interesting conclusions regarding the importance weights of features were provided.

**Author Contributions:** Conceptualization, B.L.; Data curation, G.L. and Y.L.; Formal analysis, B.L.; Funding acquisition, J.Y. and H.Z.; Methodology, B.L.; Resources, G.L. and Y.L.; Software, B.L. and Z.Z.; Validation, H.Z.; Visualization, Q.Y. and Z.Z.; Writing—original draft, B.L.; Writing—review and editing, J.Y. and N.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** The work was partially supported by National Key R&D Program of China (2017YFB1401000) and NSFC (U1609220, 61672231, U19B2035, 61972250).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Yan, J.; Zhang, C.; Zha, H.; Gong, M.; Sun, C.; Huang, J.; Chu, S.; Yang, X. On machine learning towards predictive sales pipeline analytics. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
2. Metzger, A.; Leitner, P.; Ivanović, D.; Schmieders, E.; Franklin, R.; Carro, M.; Dustdar, S.; Pohl, K. Comparing and combining predictive business process monitoring techniques. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, *45*, 276–290. [[CrossRef](#)]
3. Wang, W.; Zhang, W.; Wang, J.; Yan, J.; Zha, H. Learning sequential correlation for user generated textual content popularity prediction. In Proceedings of the 2018 International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; pp. 1625–1631.

4. Xiao, S.; Yan, J.; Li, C.; Jin, B.; Wang, X.; Yang, X.; Chu, S.M.; Zha, H. On modeling and predicting individual paper citation count over time. In Proceedings of the 2016 International Joint Conference on Artificial Intelligence (IJCAI), New York, NY, USA, 9–15 July 2016; pp. 2676–2682.
5. Liu, X.; Yan, J.; Xiao, S.; Wang, X.; Zha, H.; Chu, S.M. On predictive patent valuation: Forecasting patent citations and their types. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–10 February 2017.
6. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
7. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations (ICLR) 2015, San Diego, CA, USA, 7–9 May 2015.
8. Becker, J.; Breuker, D.; Delfmann, P.; Matzner, M. Designing and implementing a framework for event-based predictive modelling of business processes. In *Enterprise Modelling and Information Systems Architectures-EMISA 2014*; Gesellschaft für Informatik eV: Bonn, Germany, 2014.
9. Moon, T.K. The expectation-maximization algorithm. *IEEE Signal Process. Mag.* **1996**, *13*, 47–60. [[CrossRef](#)]
10. Breuker, D.; Matzner, M.; Delfmann, P.; Becker, J. Comprehensible Predictive Models for Business Processes. *MIS Q.* **2016**, *40*, 1009–1034. [[CrossRef](#)]
11. Chater, N.; Manning, C.D. Probabilistic models of language processing and acquisition. *Trends Cogn. Sci.* **2006**, *10*, 335–344. [[CrossRef](#)] [[PubMed](#)]
12. De Weerd, J.; De Backer, M.; Vanthienen, J.; Baesens, B. A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. *Inf. Syst.* **2012**, *37*, 654–676. [[CrossRef](#)]
13. Folino, F.; Guarascio, M.; Pontieri, L. Discovering context-aware models for predicting business process performances. In Proceedings of the OTM Confederated International Conferences “On the Move to Meaningful Internet Systems”, Rome, Italy, 10–14 September 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 287–304.
14. Li, L.; Yan, J.; Yang, X.; Jin, Y. Learning interpretable deep state space model for probabilistic time series forecasting. In Proceedings of the 2019 International Joint Conference on Artificial Intelligence (IJCAI), Macao, China, 10–16 August 2019; pp. 2901–2908.
15. Quinlan, J.R. *C4. 5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.
16. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
17. Friedman, N.; Geiger, D.; Goldszmidt, M. Bayesian network classifiers. *Mach. Learn.* **1997**, *29*, 131–163. [[CrossRef](#)]
18. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 344.
19. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [[CrossRef](#)]
20. Mikolov, T.; Karafiát, M.; Burget, L.; Černocký, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Chiba, Japan, 26–30 September 2010.
21. Sundermeyer, M.; Schlüter, R.; Ney, H. LSTM neural networks for language modeling. In Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association, Portland, OR, USA, 9–13 September 2012.
22. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
23. Li, L.; Yan, J.; Wang, H.; Jin, Y. Anomaly Detection of Time Series With Smoothness-Inducing Sequential Variational Auto-Encoder. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, doi:10.1109/TNNLS.2020.2980749. [[CrossRef](#)] [[PubMed](#)]
24. Wang, G.; Qin, Z.; Yan, J.; Jiang, L. Learning to select elements for graphic design. In Proceedings of the 2020 International Conference on Multimedia Retrieval, Dublin, Ireland, 8–11 June 2020; pp. 91–99.
25. Evermann, J.; Rehse, J.R.; Fettke, P. A deep learning approach for predicting process behaviour at runtime. In Proceedings of the 2016 International Conference on Business Process Management, Rio de Janeiro, Brazil, 18–22 September 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 327–338.

26. Tax, N.; Verenich, I.; La Rosa, M.; Dumas, M. Predictive business process monitoring with LSTM neural networks. In Proceedings of the 2017 International Conference on Advanced Information Systems Engineering, Essen, Germany, 12–16 June 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 477–492.
27. Polato, M.; Sperduti, A.; Burattin, A.; de Leoni, M. Time and activity sequence prediction of business process instances. *Computing* **2018**, *100*, 1005–1031. [[CrossRef](#)]
28. Rogge-Solti, A.; Weske, M. Prediction of remaining service execution time using stochastic petri nets with arbitrary firing delays. In Proceedings of the 2013 International Conference on Service-Oriented Computing, Berlin, Germany, 2–5 December 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 389–403.
29. Maggi, F.M.; Di Francescomarino, C.; Dumas, M.; Ghidini, C. Predictive monitoring of business processes. In Proceedings of the 2014 International Conference on Advanced Information Systems Engineering, Thessaloniki, Greece, 16–20 June 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 457–472.
30. Yan, J.; Liu, X.; Shi, L.; Li, C.; Zha, H. Improving maximum likelihood estimation of temporal point process via discriminative and adversarial learning. In Proceedings of the 2018 International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; pp. 2948–2954.
31. Wu, W.; Yan, J.; Yang, X.; Zha, H. Reinforcement Learning with Policy Mixture Model for Temporal Point Processes Clustering. *arXiv* **2019**, arXiv:1905.12345.
32. Xiao, S.; Farajtabar, M.; Ye, X.; Yan, J.; Song, L.; Zha, H. Wasserstein learning of deep generative point process models. In Proceedings of the Advances in Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 3247–3257.
33. Wu, W.; Yan, J.; Yang, X.; Zha, H. Decoupled learning for factorial marked temporal point processes. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2516–2525.
34. Xiao, S.; Yan, J.; Farajtabar, M.; Song, L.; Yang, X.; Zha, H. Learning time series associated event sequences with recurrent point process networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3124–3136. [[CrossRef](#)] [[PubMed](#)]
35. Yan, J.; Xu, H.; Li, L. Modeling and applications for temporal point processes. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; ACM: New York, NY, USA, 2019; pp. 3227–3228.
36. Xiao, S.; Xu, H.; Yan, J.; Farajtabar, M.; Yang, X.; Song, L.; Zha, H. Learning conditional generative models for temporal point processes. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
37. Xiao, S.; Yan, J.; Yang, X.; Zha, H.; Chu, S.M. Modeling the intensity function of point process via recurrent neural networks. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
38. Wu, Q.; Zhang, Z.; Gao, X.; Yan, J.; Chen, G. Learning latent process from high-dimensional event sequences via efficient sampling. In Proceedings of the 2019 Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 3847–3856.
39. Wu, W.; Yan, J.; Yang, X.; Zha, H. Discovering Temporal Patterns for Event Sequence Clustering via Policy Mixture Model. *IEEE Trans. Knowl. Data Eng.* **2020**, doi:10.1109/TKDE.2020.2986206. [[CrossRef](#)]
40. Wang, C.; Han, D.; Liu, Q.; Luo, S. A deep learning approach for credit scoring of Peer-to-Peer lending using attention mechanism LSTM. *IEEE Access* **2018**, *7*, 2161–2168. [[CrossRef](#)]
41. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 2013 Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 3111–3119.
42. Fu, X.; Zhang, S.; Chen, J.; Ouyang, T.; Wu, J. A Sentiment-Aware Trading Volume Prediction Model for P2P Market Using LSTM. *IEEE Access* **2019**, *7*, 81934–81944. [[CrossRef](#)]
43. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
44. Kerber, R. Chimerge: Discretization of numeric attributes. In Proceedings of the Tenth National Conference on Artificial Intelligence, San Jose, CA, USA, 12–16 July 1992; pp. 123–128.

45. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. In Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
46. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 785–794.
47. Rendle, S. Factorization machines. In Proceedings of the IEEE International Conference on Data Mining, Vancouver, BC, Canada, 11–14 December 2011.

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).