



Customer churn prediction for a webcast platform via a voting-based ensemble learning model with Nelder-Mead optimizer

Kani Fu¹ · Guiyang Zheng¹ · Wei Xie¹

Received: 21 March 2023 / Revised: 7 June 2023 / Accepted: 8 June 2023 /

Published online: 30 June 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

This article studies the application for customer churn prediction on webcast. Predicting churn customers become an urgent need in webcast industry because the market is getting saturated and identifying potential churn customers and developing recall marketing strategies can save companies significant costs. Despite the importance of customer churn prediction in many fields, little prior academic attention has been attached to the webcast area. To address this gap, We apply an ensemble learning method to build a binary classification model for customer churn prediction. Our proposed model uses a weighted voting ensemble method and the Nelder-Mead optimal algorithm with a specific focus on the speed of Internet customers' mobility, extracting high-dimensional features from time series data to incorporate more detailed customer behavior information. In addition, a new customer churn indicator based on time decline is introduced to more accurately define churned customers in the training data. The experimental data is collected from a webcast application developed by a Chinese Internet company. Experimental evaluations show that compared to the traditional ensemble models, our proposed model is operationally efficient and outperforms other approaches, providing valuable insights for companies to intervene with churned customers and adopt targeting retention interventions.

Keywords Customer churn prediction · Ensemble learning model · Machine learning · Webcast · Time-series feature · Nelder-Mead simplex algorithm

Guiyang Zheng and Wei Xie contributed equally to this work.

✉ Wei Xie
bmwxie@scut.edu.cn

Kani Fu
karlie330.f@gmail.com

Guiyang Zheng
Wanna_FlyYoung@outlook.com

¹ School of Business Administration, South China University of Technology, Guangzhou 510640, Guangdong, China

1 Introduction

Customer churn prediction has been an important area of research in customer relationship management. Since the cost of retaining existing customers is five to six times lower than the cost of attracting new ones (Rosenberg and Czepiel, 1984), identifying potential churn customers, especially regular customers and developing recall marketing strategies for them, can result in large cost savings. Over the last decades, various attempts have been made to predict customer churn in many competitive fields, including telecommunications, financial service, retail, e-commerce, and others (Verbeke et al., 2012; Tang et al., 2014; HsinChu et al., 2017; Gordini and Veglio, 2017). With the rapid development of the Internet industry in recent years, the number of Internet applications has increased dramatically. However, as Internet penetration rises, the industry's growth is slowing. As a result, the Internet application market, including the webcast market, is becoming increasingly saturated. According to a report by China Internet Network Information Center, the webcast industry in China has seen a slowdown in growth, with the user scale and usage rate decreasing in the six months leading up to June 2022.¹ Therefore, there is an urgent need to develop effective customer churn prediction strategies in the webcast industry.

Customer churn prediction in the webcast area has, however, attracted comparatively little prior academic attention. A webcast application is a platform that enables users to create rooms and share live video or audio content with viewers. Audiences in webcast rooms can engage with the content through commenting and talking, with the permission of the room creator. In some applications, audiences are incentivized to acknowledge high-quality content through a reward system, which ensures the quality of the platform and enhances its revenue. Customers who quit the platform and stop using the application are known as churners in the webcasting industry. Unlike in contract-based businesses where customers are considered churned when the contract is terminated, identifying churned customers can be difficult in industries that lack strong contractual relationships. This type of relationship implies that it is the customer's personal decision to churn (Wang et al., 2019), which makes it difficult for companies to accurately label churners. Companies that fail to accurately identify customers who are likely to churn and take action will eventually lose those customers. Moreover, mistakenly assuming that loyal customers are unlikely to leave can result in wasted operational costs. Therefore, it is important that we develop a reliable strategy for predicting customer churn in the webcast area.

Based on prior literature, artificial intelligence techniques have been extensively applied for customer churn prediction research. Some researchers have explored data-driven approach in recommendation systems for improving customer loyalty (Tarnowska et al., 2020). The utilization of knowledge-based approaches, such as sentiment analysis and action rule mining, has shown significant effectiveness in improving customer churn rate. (Duan and Ras, 2022; Tarnowska and Ras, 2021) Also machine learning-based techniques have been widely used in several fields (Rahman and Kumar, 2020; Lalwani et al., 2022; Xiahou and Harada, 2022; Amin et al., 2019). Ensemble learning is one of the most efficient applications of machine learning, as it combines multiple models into a single model, resulting in increased accuracy and reduced model variance (Sagi and Rokach, 2018). According to data from various research, combining multiple learning models produces better results than using just one learner (Zhang and Ma, 2012). So ensemble models have gained popularity in various machine learning applications and cognitive science research over the past decade. Given the nature of customer churn prediction, binary classification models are typically used to

¹ <https://cnnic.cn/n4/2022/0916/c38-10594.html>

distinguish customers likely to churn (positive class) from those likely to continue using the webcast application (negative class). For customer churn prediction this article about to explore, the classifier ensemble model will be used.

Constructing an ensemble classifier model involves two fundamental steps: selecting a training methodology for base classifiers and determining a suitable approach to combine their outputs. There are several existing methods for combining base classifiers' outputs, including majority voting, plurality voting, weighted voting and so on (Kuncheva and Rodríguez, 2014). In our work, we focus on the weighted voting ensemble method, which strategically generates and optimally combines base classifiers using different coefficients assigned to each classifier.

Various research carried out used Nelder-Mead algorithm and better solved their problems (Liu et al., 2020; Weng et al., 2021; Neshat et al., 2022). Due to the combined methods of ensemble learning, Nelder-Mead optimal algorithm can be useful. Since there is optimization space to explore for weight vector. Therefore, we aim to predict the customer churn in webcast more efficiently by improving the weighted soft voting method with Nelder-Mead optimizer. Based on the previous prediction work carried out in the related fields, this article uses data collected from a webcast application developed by a Chinese Internet company as experimental data.

The main contribution of this paper can be summarized as follows:

- The method we proposed uses NMs' idea to enhance the performance of the weighted soft voting algorithm. It improves the process of searching for the optimal weight values for model combinations.
- The effectiveness of this improved soft-voting algorithm is verified by experiments based on a real-world dataset. The results show the superior performance of our models in identifying potential churners.
- Our research contributes to the field of customer churn prediction in the webcast sector by proposing specific churn indicators, which offer new insights into the application of machine learning.
- To address the challenge of customer churn in the webcast sector, we propose an early prediction model that can identify potential churners before they leave the platform. Our model leverages machine learning techniques and can provide companies with valuable insights to prevent customer loss and increase loyalty.

This paper is structured as follows. In the next section previous work related to customer churn prediction is briefly discussed. The proposed method, including the newly defined churn indicators, the related algorithm, and the customer churn prediction model we proposed will be introduced in Section 3. In Section 4, we discuss the data definitions and preprocessing techniques used. Section 5 presents the evaluation setup and computational results. Finally, in Section 6, conclusions are drawn based on the findings of the study.

2 Related work

2.1 Churn definition

The relationship between customers and business platforms can generally be divided into two categories: contract-based platforms and registration-based platforms. In a contract-based platform, customers are provided with a service for the duration of the contract, and

the contract is terminated if the customers cancel the service. In contrast, in a registration-based platform, the basic service is available as long as customers have registered an account, and customers can stop using the platform without performing any operations. In previous literature, it has been observed that in contract-based platforms, such as those in the telecommunications and financial services industries, most researchers label a customer as a churner when they terminate their contract (Huang et al., 2012; Mena et al., 2019).

To make recall strategies more effective, some studies, however, focus on contract-based platforms and start to predict churners before they leave by referring to the churn definition approach of registration-based platforms. Alboukaey et al. (2020) conduct research in mobile telecom. They define a customer as a churner that stops doing revenue-generating events during the next 30 days because they found 81% of them have zero activity during the last thirty days of the observation period.

For most customer churn prediction studies in the registration-based platform which is basically the Internet industry proposed, they determine customer churn by judging if the number of consecutive days of customer inactivity exceeds a preset threshold. Wang et al. (2019) define a customer as a churner if the one has no clicks in 90 days in the search advertising market. However, this method seems to be too general. Because the relationships between the platform and customers are varied in different fields. To solve this problem, Alberts et al. (2006) proposed an alternative definition of customer churn named Operational Churn. It involves two parts α and β , where α is a fixed value determined by a specific application sector and β denotes the maximum consecutive inactivity days of customers in the past. When the number of consecutive inactivity days of the customer is larger than $\alpha + \beta$, this customer is labeled as a churner. Although Alberts' method improves the shortage of the traditional method to a certain extent, it still can't properly describe the customer churn behavior in the Internet application field, where every customer has very much different usage frequency. All churn indicator definitions mentioned above need a long observation period, which is not helpful for companies to retain soon-to-be-lost customers. Because when the prediction process has finished, customers already leave the platform. Considering this, we proposed a new churn indicator in Section 3.1 to improve the definition of churn customers among previous research.

2.2 Customer churn prediction using machine learning

During the last ten years, customer churn prediction has received increasing attention, especially using machine learning methods. Several types of Machine Learning (ML) algorithms have been applied to the problem of churn prediction, including decision trees, logistic regression, random forest, neural networks, and ensemble learning.

Much of the early research on customer churn prediction using machine learning applied conventional classification methods such as decision trees and logistic regression. Burez and Van den Poel (2007) used logistic regression with Markov chains for the selection of easy-to-churn customers.

To guarantee the accuracy and generalization of the constructed models for large-scale, nonlinearity, and high-dimensional, there has been a growing interest in using neural networks. A study, published by Mena et al. (2019) used LSTM neural networks combined with RFM variables to predict churn for a financial services provider from Europe. They found that the LSTM model with RFM variables has a larger top-decile lift than regularized logistic regression models with commonly used demographic variables.

Another approach has been to use ensemble learning techniques, which involve training multiple ML models and combining their predictions to make a final prediction. Liu et al. (2022) introduced ensemble learning into the telecom customer churn field to improve the effectiveness and robustness of the model. Lemmens and Croux (2006) explored the bagging and boosting classification techniques and both significantly improved accuracy in prediction churn. A study by De Caigny et al. (2021) proposed a new hybrid algorithm named LLM based on logistic regression and decision trees for B2B customer churn prediction. This new approach has performed better than its building parts logistic regression and decision trees and at least as well as more advanced ensemble methods random forest and logistic model trees.

Furthermore, there have been several additional research efforts aimed at improving the accuracy of churn prediction by applying new forms of features. One method has been to incorporate additional features into the data. A study by Huang et al. (2012) presented a new set of features for land-line customer churn prediction and the experimental results showed that the new features are more effective than the existing ones. Another method has been to build models based on dynamic features at different time levels. For example, Óskarsdóttir et al. (2018) proposed a novel method to extract time-series data from call networks to represent dynamic customer behavior at a weekly level and use the similarity forests method together with some of the proposed extensions to predict churn. While Ahmad et al. (2019) used customer social network analysis (SNA) to build features at a monthly level, and the results showed that the use of SNA enhanced the performance of the model from 84% to 93.3% against the AUC standard.

In conclusion, customer churn prediction using machine learning methods has proven to be valuable for companies looking to proactively prevent customer churn and improve customer retention. Through previous literature, there is increasing research focus on the construction of time series features to respond flexibly to market changes, which is largely dependent on the quality and relevance of the data. However, only a few studies have focused on churn prediction in the scenario of webcast. As a result, our research can fill up gaps in the field.

3 Proposed method

3.1 Churn indicator

In this section, we propose a new customer churn indicator that considers the decline rate of a customer's usage time. We use this indicator to segment lost customers from existing data and name it the Modified-Time-Decline (MTD) indicator. By detecting changes in a customer's current and previous usage duration, we can predict customer churn at an early stage, allowing the company to take timely actions to retain customers. Specifically, we define t_1 and t_2 as the customer activity time of two consecutive time windows. Then $\delta = \frac{t_2 - t_1}{t_1}$ denotes the rate of decrease in the customer's activity in the two windows. d_1 means the customer's active days in the first time window. So the proposed churn indicator can be constructed as follow:

$$I_{MTD} = e^{-(d_1-1)/k} \cdot \frac{t_2 - t_1}{t_1} \quad (1)$$

where k is a hyperparameter and $e^{-(d_1-1)/k}$ is a correction term considering the higher returning rate of customers who are more active within the observation window. According

to the churn determination, customers with a churn indicator above a certain threshold value are defined as churners. The churn determination rule is defined as:

$$\text{Label}(c_i = \text{churner}) = \begin{cases} 1, & I_{MTD} > T \\ 0, & I_{MTD} \leq T \end{cases}, \quad i \in Z^+, \quad (2)$$

where T is the threshold of the proposed churn indicator. When the value of indicator I_{MTD} is larger than the threshold, customer c_i is labeled as a churner. Otherwise, s/he is not.

In Section 4.4, an experiment will be conducted to ascertain the superiority of the I_{MTD} as a churn indicator, compared to the Operational Churn indicator (I_{OC}) and Unmodified Time-Decline indicator (I_{TD}). And also, a reasonable threshold T will be set by the experiment. The chosen churn indicator will be subsequently employed to identify churn customers.

3.2 Soft voting algorithm

In the soft voting algorithm, each base learner outputs a probability score for each class, and these scores are constructed as a score vector (Tasci et al., 2021). Instead of directly using class labels for classifier ensemble, the probability score-vector is used for vote aggregation under a combination rule, such as averaging or weighted averaging of probabilities. Given a set of N classifiers, h_1, h_2, \dots, h_T , and an input sample x , the soft voting algorithm predicts the class label $H(x)$ by taking the class with the highest probability score p_c as follows:

$$H(x) = \arg \max_c \sum_{i=1}^T \omega_i p_{c,i} \quad (3)$$

$$\sum_{i=1}^T \omega_i = 1, \quad \omega_i \geq 0$$

where $p_{c,i}$ is the probability score of class c output by classifier h_i . The final prediction is made by combining the probability scores from all base classifiers using the weighted sum (3). The weights ω_i can be adjusted to emphasize the contribution of base learners with higher performance or to downplay the contribution of those with lower performance. This approach provides a more flexible mechanism to highlight the strengths of base learners who have good performance while mitigating the impact of those who perform poorly. By incorporating the probability scores of base learners, the soft voting algorithm can achieve better performance than the hard voting algorithm, which simply counts the votes of base learners to determine the final prediction.

3.3 Nelder-Mead simplex method

The Nelder-Mead algorithm, first proposed by Jone Nelder and Roger Mead in 1965, is widely used for minimizing functions of several variables (Singer and Nelder, 2009). It operates by comparing function values at $n + 1$ vertices of a constructed simplex in the search space. Specifically, the algorithm selects $n + 1$ vertices to construct a polyhedron in an n -dimensional search space, calculates the function value at each of the $n + 1$ vertices, and sorts the function values. The algorithm then changes the vertices that have the worst function values in order to find a better solution, using operations such as reflection, expansion, and compression as showed in Fig. 1. The specific process of the Nelder-Mead algorithm can be described as follows:

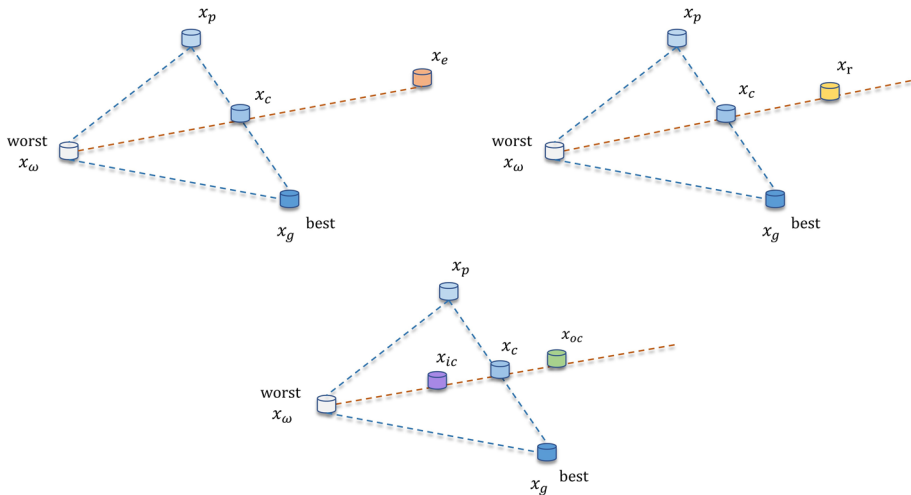


Fig. 1 Possible search behavior of the Nelder-Mead algorithm

First, we initialize $n + 1$ affine independence points as vertices of the simplex in n -dimensional space. Calculate the function values of all vertices to find the best one x_g , the second one x_p and the worst one x_w . The corresponding function values are $f(x_g)$ and $f(x_p)$, and the center position of these two is:

$$x_c = (x_g + x_p)/2 \tag{4}$$

Then, construct reflection point x_r as (5), where α denotes the reflection coefficient.

$$x_r = x_c + \alpha(x_c - x_w) \tag{5}$$

Expansion operation: if $f(x_r) < f(x_g)$, the reflection direction toward the lowest point is correct. So the expansion operation is performed to move the point to a promising solution set. In (6) we can obtain the extension point x_e , where β denotes the extension factor. If $f(x_e) < f(x_g)$, then replace x_w by x_e , make $x_w = x_e$, otherwise replace x_w by x_r .

$$x_e = x_c + \beta(x_r - x_c) \tag{6}$$

Compression operation: if $f(x_r) > f(x_g)$, the reflection direction toward the lowest point is wrong. So compression operation will be performed. According to (7), the compression point x_{ic} is obtained, where γ is the compression coefficient. If $f(x_{ic}) < f(x_w)$, then let $x_w = x_{ic}$.

$$x_{ic} = x_c + \gamma(x_r - x_c) \tag{7}$$

In (8) we can obtain the contraction point x_{oc} , where τ is the contraction coefficient, which has the same value as compression coefficient γ . If $f(x_{oc}) < f(x_w)$, let $x_w = x_r$

$$x_{oc} = x_c - \tau(x_w - x_c) \tag{8}$$

3.4 The proposed Nelder-Mead optimization-based weighted voting

We introduce NMs method into soft-voting algorithm to improve weights assigning for submodels so that the final optimal model built is of higher accuracy. The overview of the

proposed NM-SoftVoting method that utilize NMs method and soft-voting is shown in Fig. 2.

The method consists of two phases. Beforehand, we divide the dataset into two datasets: training set $D = \{x_i, y_i\}_{i=1}^m$ and testing set $S = \{x_j, y_j\}_{j=1}^n$, where x and y denote feature and label respectively. In the first phase, a certain percentage of the training set data is selected as a leave-out set, and the remaining data in the training set are used to train base learning models. These base learners are then used to predict the data from the leave-out set and the test set. The process of the first phase is outlined below:

Training set D is divided into two sets, a new training set $B = \{x_i, y_i\}_{i=1}^a$ and a new leave-out set $H = \{x_i, y_i\}_{i=1}^b$.

- (1) T base learning models are trained using the new training set B , and thus T strong learners h_1, h_2, \dots, h_T are obtained.
- (2) Strong learners obtained in step 2 are used to predict features in the leave-out set H . After predicting, an outcome set including features x_i and prediction value \bar{y}_i' is obtain, denoted as $H_h = \{x_i, \bar{y}_i'\}_{i=1}^b$, where $\bar{y}_i' = \{h_1(x_i), h_2(x_i), \dots, h_T(x_i)\}$.
- (3) Similarly, using strong learners to predict features in the testing set S . Then $S_h = \{x_j, \bar{y}_j'\}_{j=1}^n$ is obtained, where x_j denotes feature, \bar{y}_j' denotes prediction value and $\bar{y}_j' = \{h_1(x_j), h_2(x_j), \dots, h_T(x_j)\}$.

In the second phase, we improve the soft-voting algorithm searching strategy using the NMs method, such that the voting outcome based on advanced weights can have a better

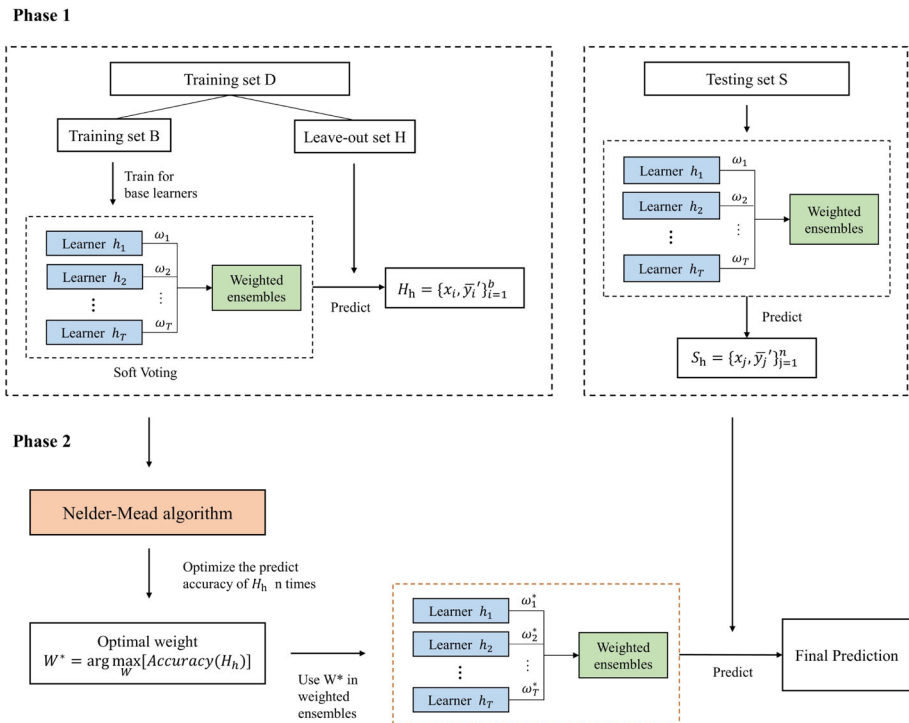


Fig. 2 Framework of NM-SoftVoting algorithm

performance. In the voting process, the prediction value of feature x_i under weight W can be defined as:

$$\langle \overline{y}_i', W^T \rangle = [h_1(x_i) \ h_2(x_i) \ \cdots \ h_T(x_i)] \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_T \end{bmatrix} = s_i. \tag{9}$$

s_i in (9) represents that customer i has a probability s_i of being a churned customer. Because the problem we solve is a binary classification problem, we set the final prediction label as “1” when the prediction value exceeds 0.5. Otherwise, it is “0”. Therefore, the final prediction result for feature x_i is:

$$\overline{y}_i = \begin{cases} 1, & s_i > 0.5 \\ 0, & s_i \leq 0.5 \end{cases} \tag{10}$$

Through the performance matrix, we can calculate the corresponding accuracy rate, recall rate, and F1 value. It is easy to know that there is a certain functional relationship between accuracy rate and weights W . So the function of accuracy about the weights $f(W)$ can be set as the objection function for optimal weighted value searching. Then NMs algorithm is used here to optimize the function whose specific form is unknown and is not derivable. Like most direct search methods, NMs easily falls into local extremes, so we will perform the algorithm multiple times based on different initial simplexes. The specific process of the second phase is described as followed:

- (1) Applying the Nelder-Mead method on the weights assigning process for the pre-train base learners to find the best weights W^* .
- (2) Performing the Nelder-Mead method on H_h multiple times based on different initial simplexes to avoid falling into local extremes.
- (3) Comparing these model performances and choosing the weights of the best model as a final weighted number.
- (4) Applying pre-train models in phase one and the best weights searched on the testing set S and getting the final testing performance.

The choice of the initial simplex is significantly important since simplexes are too small and are also more likely to fall into local extremes. Here we use the way mentioned in the Nelder-Mead literature to construct the simplex. Given an initial point $W_0 = (\omega_1, \omega_2, \dots, \omega_T)$, the remaining T points are obtained in different dimensional directions with a fixed step δ , together these points form the initial simplex.

$$\begin{bmatrix} W_0 \\ W_1 \\ \vdots \\ W_T \end{bmatrix} = \begin{bmatrix} \omega_0 & \omega_1 & \cdots & \omega_T \\ (1 + \delta)\omega_0 & \omega_1 & \cdots & \omega_T \\ \vdots & \vdots & \ddots & \vdots \\ \omega_0 & \omega_1 & \cdots & (1 + \delta)\omega_T \end{bmatrix} \tag{11}$$

The initial point W_0 is approximately Dirichlet Distribution with the probability density function as follows:

$$f(x_1, \dots, x_T; \alpha_1, \dots, \alpha_T) = \frac{1}{B(\alpha)} \prod_{i=1}^T x_i^{\alpha_i-1} \tag{12}$$

$$B(\alpha) = \frac{\prod_{i=1}^T \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^T \alpha_i)}, \alpha = (\alpha_1, \dots, \alpha_T) \tag{13}$$

So W_0 meets $\sum_i \omega_i = 1$, which is the sum of the initial weight of each classifier equals 1. In the meantime, to make the initial points generated at the start of each cycle more random and to improve the global optimum’s search ability. In the Dillikere distribution, the hyperparameters are set as $\alpha_i = 1, i = 1, \dots, T$.

It should be noted that the solutions obtained by Nelder-Mead during the search process and the final weight sum may not necessarily add up to 1. However, this does not affect the final results since the classification is determined using a fixed threshold of 0.5 for positive and negative samples. Even if the weighted sum is not 1, it can be normalized and the threshold value adjusted accordingly. At this point, the algorithm adjusts not only the weights of the various base learners but also the threshold value used for the final decision.

4 Data definitions and preprocessing

4.1 Data description

We collected actual customer data for our research via a webcast module in a social gaming application developed by a leading webcast company. Due to confidential purpose, the real dataset used in our research cannot be disclosed. Our dataset includes customers’ static and dynamic behavior from September 18th, 2020, to November 17th, 2020.

We divided the 60-day time sequence into three time windows, namely the analysis window, the observation window, and the prediction window, as illustrated in Fig. 3. The analysis window corresponds to the first 30 days of the 60-day window, and it was used to construct the multivariate time series dataset described in Section 4.2. The observation window covers the last 15 days of the analysis window and was used to observe the decline in customer usage time to construct the I_{MTD} , which is explained in Section 3.1. The prediction window is the 30 days following the analysis window, during which we utilized our proposed method in Section 3.4 to predict whether customers would churn or not. To ensure data completeness and stability of behavior patterns, we filtered out customers who had registered less than 14 days prior to the start of the observation period, as they were not our targeted audience for retention efforts. After filtering, we were left with a total of 219,910 observations.

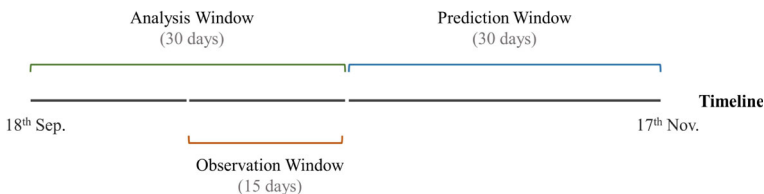


Fig. 3 Time windows for churn definition in webcast

4.2 Feature construction

We categorized the data available from the customer behavior information we collect into four groups: usage records, gift and monetary records, social attribute records, and demographic information. These categories are described as follows:

Among the four data categories outlined in Table 1, the first category, usage records, consists of dynamic time series data exhibiting changes daily. Conversely, the last three categories, namely gift and monetary records, social attribute records, and demographic information are static data. The basic feature data is derived from two distinct sources. One component is derived from static data, while the other is calculated using the daily usage time series data. In addition, we extract additional information from customer's daily active time series to generate high-dimensional time series features, which has not been well considered in previous studies.

The high-dimensional time series features include:

- Basic statistical information about the distribution of time series, including the series' distribution type, divergence, Gaussianity values, outlier properties, etc.
- Linear correlation, including autocorrelation, power spectral density, etc.
- Stationarity, including StatAv, sliding window measures, prediction errors, etc.
- Information theory and complexity measures, including self-information and mutual information, approximate entropy, and Lempel-Ziv complexity.
- Linear and nonlinear model fitting, including ARMA, Gaussian process, and GARCH model.

Research in (Christ et al., 2017) has shown that the feature extraction on basis of scalable hypothesis tests (FRESH) method outperforms traditional methods such as time-series similarity-based algorithms and feature extraction algorithms in terms of prediction accuracy and computational speed. So FRESH framework is used to extract high-dimensional features from customers' daily usage records within a chosen time window.

To construct the input data matrix for our method, it is necessary to extract a feature vector indicating customer behavior for each day. Feature selection methods can be categorized into three types: filter, wrapper, and embed (Chandrashekar and Sahin, 2014). We have employed the filter method and used different hypothesis testing methods depending on the type of

Table 1 Description of data categories

Data Categories	Description
Usage records	Daily aggregated data on customer behavior, such as active time, time using the microphone, room entry count, etc. for a set period.
Gift and monetary records	Gift and monetary records including quantity and value of gifts sent/received, max recharge amount, etc. count at the end of Analysis Window.
Social attribute records	customer social attribute data for a set period, such as following people, friends, and followers, etc. count at the end of Analysis Window.
Demographic information	demographic information of customers about their country, age, gender, number of days since registration, etc.

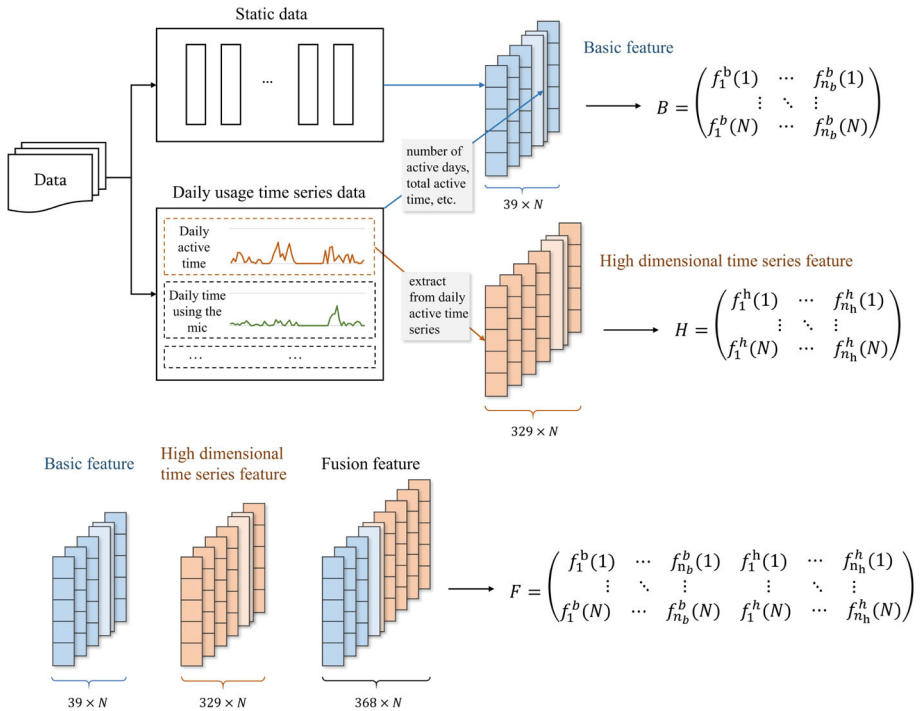


Fig. 4 Feature construction process

feature. Specifically, we extracted 39 basic features and 329 high-dimensional features for constructing the input data matrix.

The feature construction process is shown in Fig. 4. In Section 5, we will conduct experiments with different types of feature forms to explore a better optimal method for our proposed model. So basic feature, high dimensional time series (HDTs) feature, and fusion feature combined with the basic feature and HDTs feature will be tested in experiments.

4.3 Churn indicator evaluation metrics

In platform-based industries, the line between churned and non-churned customers can be vague, making it necessary to define churn definition evaluation metrics to better compare the performance of different churn indicators for data preprocessing. Two basic evaluation metrics are used, namely Return Rate (RR) and Omission Rate (OR). The Return Rate represents the probability of non-churned customers being incorrectly labeled as churned. On the other hand, the Omission Rate represents the probability of churned customers being incorrectly labeled as non-churned, which includes the targeted customers that are of great importance to the platform. These metrics are expressed as follows:

$$RR = \frac{\{c_i(\text{Label} = \text{churner})\} \cap \{c_k(\text{State} = \text{active})\}}{\{c_i(\text{Label} = \text{churner})\}} \quad (14)$$

$i, k \in Z^+$

$$OR = 1 - \frac{\{c_j(\text{Label} = \text{nonchurner}) \cap \{c_k(\text{State} = \text{active})\}}{\{c_j(\text{Label} = \text{nonchurner})\}} \quad j, k \in Z^+ \tag{15}$$

In *RR* and *OR*, $\{c_i(\text{Label} = \text{churner})\}$ and $\{c_j(\text{Label} = \text{nonchurner})\}$ denote the set of customers labeled as churned and non-churned in the observation window separately. $\{c_k(\text{State} = \text{active})\}$ denotes the set of customers who are still active in the prediction window.

Generally, the more strict the churn definition condition is, the higher accuracy churned label, which means a lower return rate. However, strict churn definition conditions can cause customers truly lost being easily labeled as non-churners, so *OR* will increase. Likewise, When the churn definition is relaxed, the return rate increases while the *OR* decreases. Low *RR* and low *OR* can't be satisfied at the same time, so we need to find a middle ground between these two metrics. Referring to the idea of generalized error definition in machine learning literature, we use the weighted average to combine the *RR* and *OR*. A specific weight ratio of 1 : 2 is chosen depending on experience in the webcast field. The combination of *RR* and *OR* compose a new metric named Error Rate (*ER*). The *ER* is presented as follows:

$$ER = \frac{1}{3}(RR + 2 * OR) \tag{16}$$

4.4 Experiment of proposed churn indicator

The purpose of this section is to evaluate the performance of our proposed churn prediction method. To accomplish this, we compare our method with the best performance of Operational Churn indicator (*IOC*) proposed by Alberts using the same dataset. In Operational Churn indicator, customer churn threshold is determined by $\alpha + \beta$, where α is a parameter and β is fixed once the dataset is chosen. To evaluate the performance of both methods, we analyze the *RR*, *OR*, and *ER* under different values of α . As shown in Fig. 5, the Error Rate based

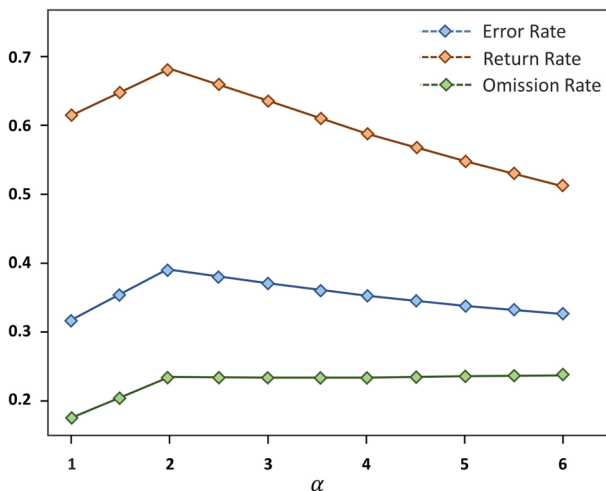


Fig. 5 Evaluation metrics of Operation Churn indicator at different α

Table 2 Evaluation metrics of Operation Churn Indicator at its best performance

Churn Definition	Return Rate	Omission Rate	Error Rate
Operational Churn	0.6158	0.1729	0.3205

on (13) is the lowest when $\alpha = 1$, and the churn threshold value of I_{OC} is $1 + \beta$. Table 2 presents the RR, OR, and ER when I_{OC} is at its best performance.

In the churn indicator we proposed, the determination threshold value, which is T in (2) is important. Thus, an experiment using different determination thresholds is implemented to analyze the churn threshold. We calculate RR and OR of a range of thresholds based on different churn indicators. I_{MTD} is defined in (1). The result shows in Fig. 6.

Figure 6 indicates that with increasing of threshold, which means the churn definition condition becomes more strict, RR decreases and OR increases. RR based on I_{MTD} is always lower than RR based on I_{TD} and they are both lower than RR based on I_{OC} when threshold exceeds a certain level. When the threshold is less than 0.8, OR based on I_{TD} and I_{MTD} are lower than OR based on I_{OC} . OR based on I_{MTD} is lower than I_{TD} when the threshold is low. And with the threshold increasing, OR based on I_{MTD} gradually surpasses OR based on I_{TD} .

Figure 7 presents the ER with different threshold based on different churn indicators. The performance of churn Indicator based on I_{MTD} is the best under all thresholds. Furthermore, the ER curve decreases fast in the threshold range of 0 to 0.3 and slowly when the threshold is greater than 0.3. So finally we choose $T = 0.3$ as churn determination threshold, and the corresponding error rate is 0.1776. After the threshold is decided, 20.51% of customers are defined as churners based on I_{MTD} among the proposed dataset.

5 Model experiment

5.1 Model experiment design

We design several model experiments for comparing the proposed model of customer churn prediction with different approaches. We first conduct eight commonly used classifiers that

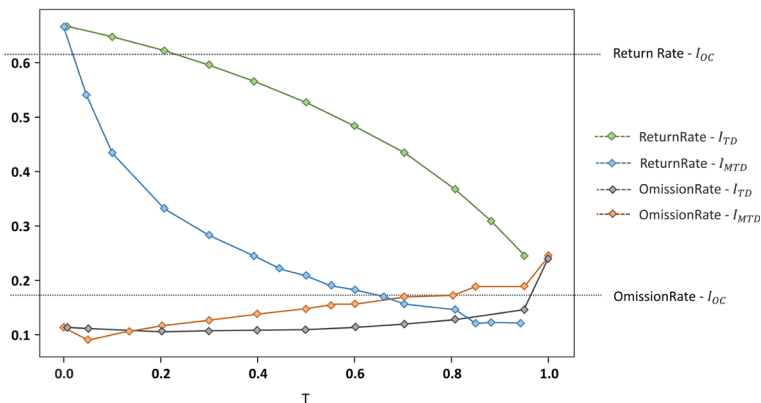


Fig. 6 Return Rate and Omission Rate curves with different thresholds based on different churn indicators

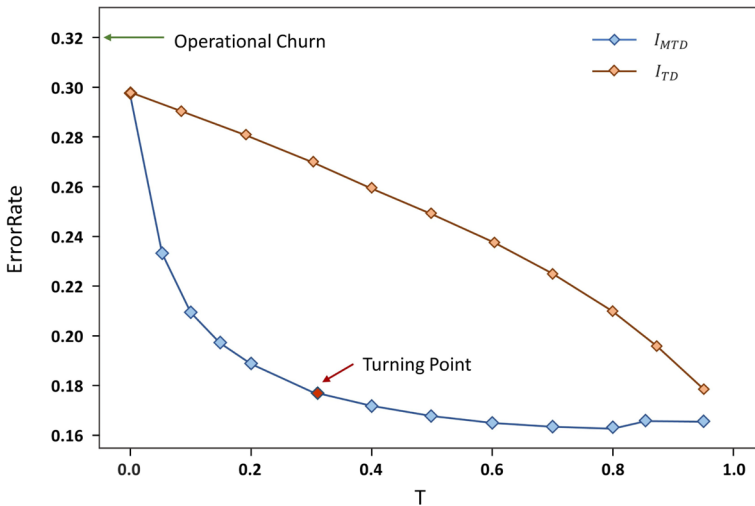


Fig. 7 Error Rate curves with different thresholds based on different churn indicators

have been widely used in the previous literature to set up a baseline. Then we select well-performed and feasible models to explore a possible better form of feature. Based on the result we obtain, our model will be compared against conventional ensemble learning methods such as Hard Voting model, Soft Voting model, Stacking model, and Blending model using superior feature form.

The following are the experiments that will be conducted:

- (1) Base classifiers including Decision Tree (DT), KNN, SVM, Back Propagation (BP), Random Forest (RF), XGBoost, LightGBM, CatBoost with basic feature.
- (2) Base classifiers well performed in our dataset and feasible using high dimensional and fusion feature.
- (3) Ensemble learning models such as Hard Voting model, Soft Voting model, Stacking model, Blending model and NM-SoftVoting model we proposed with fusion feature.

For the evaluation and comparison of models, we have used four popular performance metrics: The accuracy rate, the recall rate, the weighted harmonic mean of the precision and recall of classification (F1) and the area under the curve (AUC). Because the output form of models like SVM, Hard Voting, Blending and Stacking are non-probabilistic, their AUC of prediction can't be calculated. We will analysis the results considering other evaluation metrics comprehensively.

5.2 Result

Table 3 shows that RF is the best in accuracy, F1 value, and AUC, followed by XGBoost. We no longer use any feature reduction algorithm after we filter features in Section 4.2. Among these eight commonly used individual classifiers we presented, DT, KNN, SVM, and BP are not efficient when using high dimensional input data. So next, we choose RF, XGBoost, and two other boosting algorithms LightGBM and CatBoost, to do the comparison experiments among basic features, HDTS features, and fusion features. In Table 4, we can observe that the model based on the basic feature outperforms the model based on the HDTS

Table 3 The performance of commonly used classifiers with basic feature

Classifier	Feature Type	Accuracy	Recall	F1	AUC
DT	Basic	0.8364	0.6058	0.6015	0.7543
KNN	Basic	0.8609	0.6464	0.6545	0.9175
SVM	Basic	0.8658	0.6828	0.6748	-
BP	Basic	0.8637	0.6141	0.6469	0.9201
RF	Basic	0.8814	0.7026	0.7073	0.9398
XGBoost	Basic	0.8802	0.7040	0.7057	0.9385
LightGBM	Basic	0.8807	0.7037	0.7054	0.9390
CatBoost	Basic	0.8816	0.7016	0.7062	0.9395

feature. It implies that diverse and comprehensive basic features, such as gift and monetary records, socialization activity records, and demographic records have a greater impact on forecasting than HDTS features. Additionally, it is evident from the results depicted in Fig. 8 that models incorporating fusion feature modeling consistently outperform models with only basic features, which demonstrates the superiority of combining basic and HDTS features. This improvement is particularly prominent when considering recall rate and F1 value. To further illustrate, we use RF and XGBoost as examples. The recall rate of the two models has increased by 6.26% and 6.83%, respectively, and the F1 value has increased by 2.39% and 3.42%, indicating that the model is not only accurate in overall prediction. Simultaneously, it improves the ability to distinguish churn customers, which is exactly what the churn early warning model requires and prioritizes.

The results of the preceding experiments show that the model with the fusion feature we constructed outperforms the other models. Therefore, we will conduct experiments using different ensemble algorithms, combining four base classifiers RF, XGBoost, LightGBM, and CatBoost in the next and compare them to the performance of our proposed base classifiers combining method. In the searching process of the best weight, we take the hyperparam-

Table 4 The performance of base classifiers with three different forms of feature

Classifier	Feature Type	Accuracy	Recall	F1	AUC
RF	Basic	0.8814	0.7026	0.7073	0.9398
	HDTS	0.8610	0.6757	0.6646	0.9271
	Fusion	0.8843	0.7466	0.7242	0.9438
XGBoost	Basic	0.8802	0.7040	0.7057	0.9385
	HDTS	0.8586	0.6931	0.6666	0.9241
	Fusion	0.8864	0.7521	0.7298	0.9450
LightGBM	Basic	0.8807	0.7037	0.7054	0.9390
	HDTS	0.8589	0.6869	0.6679	0.9239
	Fusion	0.8863	0.7501	0.7290	0.9452
CatBoost	Basic	0.8816	0.7016	0.7062	0.9395
	HDTS	0.8597	0.6942	0.6645	0.9274
	Fusion	0.8856	0.7602	0.7265	0.9455

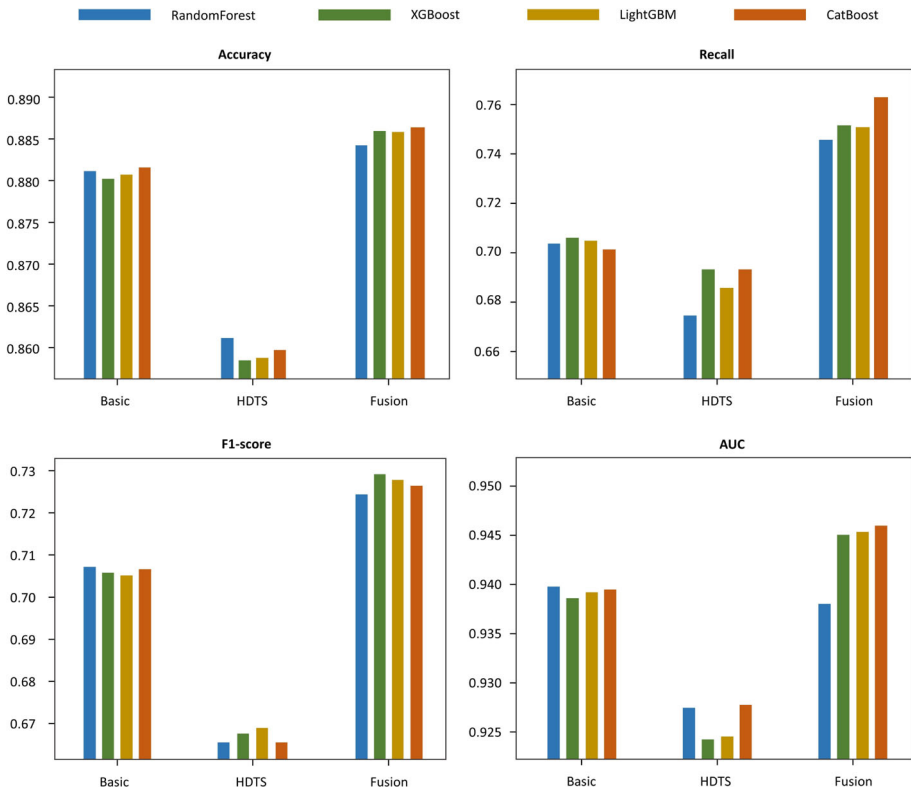


Fig. 8 The performance of base classifiers with different forms of feature

eters of our proposed model reflection coefficient α , extension coefficient β , compression coefficient γ , and compression coefficient τ standard values, which are

$$\alpha = 1, \beta = 2, \gamma = \frac{1}{2}, \tau = \frac{1}{2}.$$

The fixed step in the initial simplex construction $\delta = 0.05$. Specifically, if the initial point is 0 in a particular dimension, the point along that dimension is set to a constant value of 0.025. In addition, the number of searches K is set to 100, and the proportion of left-out set is 10% of the training sample, as in the previous Blending fusion model. Finally, the optimal weights are obtained (with four decimal places), which is $W^* = (0.3747, 0.0584, 0.5680, 0.0273)$.

Table 5 The performance of classical ensemble models and proposed model with fusion feature

Ensemble Classifier	Accuracy	Recall	F1	AUC
Hard-Voting	0.8871	0.7358	0.7266	-
Soft-Voting	0.8867	0.7496	0.7295	0.9457
Stacking	0.8849	0.7502	0.7294	-
Blending	0.8868	0.7532	0.7307	-
NM-SoftVoting	0.8867	0.7657	0.7337	0.9456

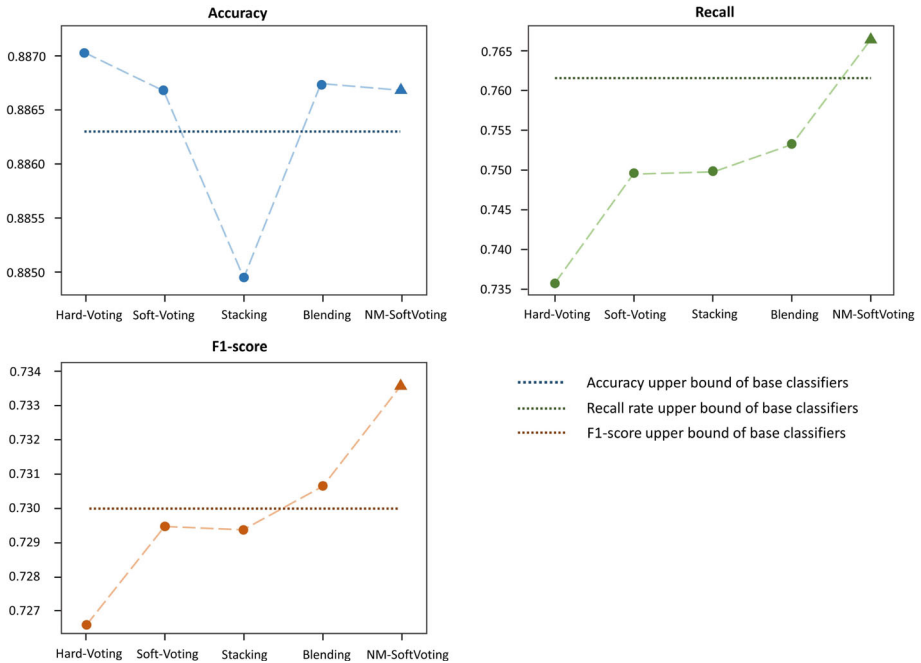


Fig. 9 The performance of classical ensemble models and proposed model with fusion feature (the horizontal dashed line is the upper bound of base classifiers)

The performance of classical ensemble models and proposed method with fusion feature are listed in Table 5. From Fig. 9 we can observe that In terms of accuracy, recall rate and F1-value, the model NM-SoftVoting we proposed outperforms the upper limit of the commonly used individual classifiers involved in our experiment. When comparing to the other four classical ensemble models, the NM-SoftVoting model has significantly better performance in terms of recall rate and F1 value when the accuracy is quite close to the best one, which is 88.67%. This means that the model improves judgmental ability for churn class samples while maintaining model’s overall accuracy. Regarding the application, an increase in recall rate indicates the model’s ability to capture churned customers has improved, which is the most important concern in a customer churn prediction model. Additionally, identifying as many churned customers as possible with high possible accuracy is a requirement for recalling more customers. Thus, our proposed model has the best performance under a integrated view.

6 Conclusion and future work

This study proposes to use Nelder-Mead optimizer to improve a voting-based ensemble learning model for the webcast industry. Our goal is to predict churn in a short period, considering the speed of Internet users’ mobility. Therefore, we extract corresponding high-dimensional features from time series data in usage frequency records to incorporate more detail of customer behavior and explore a better input feature structure, which has been

proven to be the fusion feature. Next, a new customer churn indicator based on time decline is introduced to more accurately define churned customers in our training data.

In terms of the practical experiments of this paper, results show that our proposed model is operationally efficient. Using fusion features in the same method is more effective than using other forms of features. Furthermore, ensemble learning methods have better performance in general compared with their consisting base models. With almost equal accuracy and AUC, our proposed algorithm NM-SoftVoting has a higher recall rate, which means a better ability to identify churned customers. This is very important from the business point of view.

So far, we have only considered HDTs data of customer usage information for the prediction. For future research, it would be valuable to explore the integration of dynamic social activity features. This integration has the potential to enhance the performance of the model by providing a more comprehensive view of the data. As a result, we expect an overall enhancement in the effectiveness of the model. In addition, the model is limited in its capacity to identify churn-related factors, which is necessary for companies to effectively intervene with churned customers. Further research can be conducted to identify the factors that contribute to churn and segment customers accordingly. By adopting this approach, companies can implement targeted retention interventions, thereby mitigating churn and improving customer retention. Moreover, identifying churn customers who have registered for a short time can be implemented for the integrity of the customer churn prediction problem.

Acknowledgements The authors would like to thank the Editor and two anonymous referees for their valuable comments and suggestions, which significantly improved the quality and presentation of this paper. This work was supported by National Natural Science Foundation of China [Grant 71971085], Guangzhou Basic Research Program Basic and Applied Basic Research [Grant SL2022A04J00790], and the Fundamental Research Funds for the Central Universities.

Author Contributions Conceptualization, all authors; methodology, Kani Fu, Wei Xie; experimental designs, all authors; model developing, Guiyang Zheng; original draft preparation, Kani Fu; review and editing, all authors.

Data Availability Not Applicable

Declarations

Conflict of interests The authors declare that they have no conflict of interest.

Ethics approval and consent to participate Not Applicable

Consent for publication Not Applicable

References

- Ahmad, A. K., Jafar, A., & Aljoumaa, K. (2019). Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data*, 6(1), 28. <https://doi.org/10.1186/s40537-019-0191-6>
- Alberts, L., Bsc, Peeters, R.L.M., & et al. (2006) Churn prediction in the mobile telecommunications industry an application of survival analysis in data mining master thesis
- Alboukaey, N., Joukhadar, A., & Ghneim, N. (2020). Dynamic behavior based churn prediction in mobile telecom. *Expert Systems with Applications*, 162(113), 779. <https://doi.org/10.1016/j.eswa.2020.113779>
- Amin, A., Al-Obeidat, F., Shah, B., et al. (2019). Customer churn prediction in telecommunication industry using data certainty. *Journal of Business Research*, 94, 290–301. <https://doi.org/10.1016/j.jbusres.2018.03.003>

- Burez, J., & Van den Poel, D. (2007). CRM at a pay-tv company: Using analytical models to reduce customer attrition by targeted marketing for subscription services. *Expert Systems with Applications*, 32(2), 277–288. <https://doi.org/10.1016/j.eswa.2005.11.037>
- Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), 16–28. <https://doi.org/10.1016/j.compeleceng.2013.11.024>
- Christ, M., Kempa-Liehr, A. W., & Feindt, M. (2017). Distributed and parallel time series feature extraction for industrial big data applications. <https://doi.org/10.48550/arXiv.1610.07717>. arXiv:arXiv:1610.07717
- De Caigny, A., Coussement, K., Verbeke, W., et al. (2021). Uplift modeling and its implications for B2B customer churn prediction: A segmentation-based modeling approach. *Industrial Marketing Management*, 99, 28–39. <https://doi.org/10.1016/j.indmarman.2021.10.001>
- Duan, Y., & Ras, Z. W. (2022). Recommendation system for improving churn rate based on action rules and sentiment mining. *International Journal of Data Mining, Modelling and Management*, 14(4), 287–308. <https://doi.org/10.1504/IJDDMM.2022.126665>
- Gordini, N., & Veglio, V. (2017). Customers churn prediction and marketing retention strategies. an application of support vector machines based on the auc parameter-selection technique in B2B e-commerce industry. *Industrial Marketing Management*. <https://doi.org/10.1016/j.indmarman.2016.08.003>
- HsinChu, Dingli, A., Marmara, V., & et al (2017) Comparison of deep learning algorithms to predict customer churn within a local retail industry. *International Journal of Machine Learning and Computing*, 7(5), 128–132. <https://doi.org/10.18178/ijmlc.2017.7.5.634>
- Huang, B., Kechadi, M. T., & Buckley, B. (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications*, 39(1), 1414–1425. <https://doi.org/10.1016/j.eswa.2011.08.024>
- Kuncheva, L. I., & Rodriguez, & J.J. (2014). A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems*, 38(2), 259–275. <https://doi.org/10.1007/s10115-012-0586-6>
- Lalwani, P., Mishra, M. K., Chadha, J. S., et al. (2022). Customer churn prediction system: A machine learning approach. *Computing*, 104(2), 271–294. <https://doi.org/10.1007/s00607-021-00908-y>
- Lemmens, A., & Croux, C. (2006). Bagging and boosting classification trees to predict churn. *Journal of Marketing Research*, 43(2), 276–286. <https://doi.org/10.1509/jmkr.43.2.276>
- Liu, Y., Chong, G., Heidari, A. A., et al. (2020). Horizontal and vertical crossover of harris hawk optimizer with nelder-mead simplex for parameter estimation of photovoltaic models. *Energy Conversion and Management*, 223(113), 211. <https://doi.org/10.1016/j.enconman.2020.113211>
- Liu, Y., Fan, J., Zhang, J., et al. (2022). Research on telecom customer churn prediction based on ensemble learning. *Journal of Intelligent Information Systems*. <https://doi.org/10.1007/s10844-022-00739-z>
- Mena, C.G., De Caigny, A., Coussement, K., & et al. (2019) Churn prediction with sequential data and deep neural networks. a comparative analysis. <https://doi.org/10.48550/arXiv.1909.11114>. arXiv:1909.11114
- Neshat, M., Nezhad, M. M., Sergiienko, N. Y., et al. (2022). Wave power forecasting using an effective decomposition-based convolutional bi-directional model with equilibrium nelder-mead optimiser. *Energy*, 256(124), 623. <https://doi.org/10.1016/j.energy.2022.124623>
- Óskarsdóttir, M., Van Calster, T., Baesens, B., et al. (2018). Time series for early churn detection: Using similarity based classification for dynamic networks. *Expert Systems with Applications*, 106, 55–65. <https://doi.org/10.1016/j.eswa.2018.04.003>
- Rahman, M., & Kumar, V. (2020) Machine learning based customer churn prediction in banking. In: 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp 1196–1201. <https://doi.org/10.1109/ICECA49313.2020.9297529>
- Rosenberg, L. J., & Czepiel, J. A. (1984). A marketing approach for customer retention. *Journal of Consumer Marketing*. <https://doi.org/10.1108/eb008094>
- Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249 <https://doi.org/10.1002/widm.1249>
- Singer, S., & Nelder, J. (2009). Nelder-mead algorithm. *Scholarpedia*, 4(7), 2928. <https://doi.org/10.4249/scholarpedia.2928>
- Tang, L., Thomas, L., Fletcher, M., & et al. (2014). Assessing the impact of derived behavior information on customer attrition in the financial service industry. *European Journal of Operational Research*, 236(2), 624–633. <https://doi.org/10.1016/j.ejor.2014.01.004>
- Tarnowska, K., Ras, Z. W., & Daniel, L. (2020). *Recommender System for Improving Customer Loyalty*, (Vol. 1). Springer.
- Tarnowska, K. A., & Ras, Z. (2021). Nlp-based customer loyalty improvement recommender system (clirs2). *Big Data and Cognitive Computing*, 5(1), 4. <https://doi.org/10.3390/bdcc5010004>
- Tasci, E., Uluturk, C., & Ugur, A. (2021). A voting-based ensemble deep learning method focusing on image augmentation and preprocessing variations for tuberculosis detection. *Neural Computing and Applications*, 33(22), 15541–15555. <https://doi.org/10.1007/s00521-021-06177-2>

- Verbeke, W., Dejaeger, K., Martens, D., et al. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, 218(1), 211–229. <https://doi.org/10.1016/j.ejor.2011.09.031>
- Wang, Q. F., Xu, M., & Hussain, A. (2019). Large-scale ensemble model for customer churn prediction in search ads. *Cognitive Computation*, 11(2), 262–270. <https://doi.org/10.1007/s12559-018-9608-3>
- Weng, X., Heidari, A. A., Liang, G., et al. (2021). Laplacian nelder-mead spherical evolution for parameter estimation of photovoltaic models. *Energy Conversion and Management*, 243(114), 223. <https://doi.org/10.1016/j.enconman.2021.114223>
- Xiahou, X., & Harada, Y. (2022) B2C e-commerce customer churn prediction based on k-means and svm. *Journal of Theoretical and Applied Electronic Commerce Research*, textit17(2), 458–475. <https://doi.org/10.3390/jtaer17020024>
- Zhang, C., & Ma, Y. (2012) *Ensemble Machine Learning: Methods and Applications*. Springer Science & Business Media

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.