Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 33(4): 2366–2390.
DOI: 10.3934/era.2025105
Received: 16 February 2025
Revised: 30 March 2025
Accepted: 03 April 2025
Published: 22 April 2025

Research article

Gender prediction model based on CNN-BiLSTM-attention hybrid

Wang Zichang[1,*] **and Lu Xiaoping**[2]

[1] Department of Artificial Intelligence and Data Science, Guangzhou Xinhua University, Guangdong 523133, China

[2] Computer Science and Engineering, Macau University of Science and Technology, Macau 999078, China

* **Correspondence:** Email: wzcbox@sina.cn; Tel: +8613437839282.

Abstract: Accurate gender prediction is crucial for businesses to offer personalized services to their customers. To address the issue of low prediction accuracy typically associated with traditional machine learning techniques in commercial recommendation systems, a parallel hybrid prediction model was proposed. This model combines convolutional neural networks (CNNs), bidirectional long short-term memory (BiLSTM) networks , and an attention mechanism, forming a hybrid CNN-BiLSTM-attention mechanism model. By leveraging the CNN's ability to capture local features, the BiLSTM's strength in processing the contextual information of sequential data, and the attention mechanism's focus on relevant data, the model improves gender prediction accuracy. Additionally, the research utilized the ANOVA method and random forest models to extract relevant features, applied the continuous bag of word (CBOW) algorithm to vectorize clickstream text data, and employs the parallel CNN-BiLSTM-attention mechanism model for gender prediction. The results show that this proposed model outperforms individual models in gender prediction accuracy. This development establishes a strong foundation for merchant recommendation systems, allowing them to deliver more accurate service recommendations.

Keywords: CNN; BiLSTM; hybrid parallel; attention mechanism; gender prediction

1. Introduction

In the current digital era, recommendation systems are crucial in sectors such as e-commerce, social media, and entertainment platforms. By analyzing users' past behaviors and preferences, these systems deliver personalized content suggestions, thereby enhancing user satisfaction and the overall experience. However, traditional recommendation systems primarily generate results based on browsing history, purchase records, and rating data [1–5]. While effective, this approach does not

fully leverage other potentially valuable user data. One such factor is gender, which significantly influences consumer behavior, interests, and purchasing decisions. Therefore, incorporating gender into recommendation systems can improve both the accuracy and personalization of recommendations. However, accurately determining a user's gender remains a challenge, particularly when users do not explicitly disclose this information.

By analyzing users' historical behavior and personal data, a robust gender prediction model can be developed to provide more accurate personalized recommendations. This model holds considerable practical value in areas such as targeted advertising and marketing.

2. Related work

The gender prediction system [6], based on statistical machine learning, addresses the gender prediction problem in online transnational e-commerce data by utilizing unique ID decomposition, historical generation based on a context window, and the extraction of a consistent hierarchical structure from the training set. However, this approach has some limitations. It heavily relies on the specific structure of e-commerce data, and its generalization ability to other types of data may be restricted. If the data format or business model changes, the model may require significant re-engineering.

Literature [7] proposes a method to predict user gender and age using behavior, service, and contract information. This method relies on call records, customer relationship management, and billing information to analyze telecom customer behavior and establish an accurate algorithm for demographic attributes. However, this method is constrained by its reliance on telecom-specific data sources, making it less applicable in scenarios where such telecom-related data is unavailable. Moreover, the feature extraction process is primarily focused on telecom-specific behaviors, lacking a comprehensive approach to analyze more general user behavior.

To address the challenges of age and gender classification in intelligent systems based on face images, particularly in military, criminal investigation, and visa processing fields, literature [8] proposes an enhanced framework for age prediction and gender classification. The benchmark datasets, FG-NET and HQFaces, were reduced dimensionally using partial least squares technology. A multiple regression model was employed to classify age, and a support vector machine (SVM) was used to classify gender. While the framework demonstrates superior accuracy in age prediction and gender classification compared to some other techniques, it is limited to face-image data. It cannot process non-visual data, such as text or audio, for gender prediction, narrowing its scope of application.

In literature [9], natural language processing (NLP) and machine learning methods are applied to extract various personality characteristics from text data to predict the author's age and gender. The approach integrates NLP technologies such as word segmentation, word reduction, and language modeling with logistic regression (LR), random forests, decision trees, and support vector machines, achieving good prediction results. However, this method mainly focuses on text-specific features and may not be effective at incorporating other types of user behavior data. Additionally, combining multiple algorithms can lead to high computational complexity and potential overfitting issues.

Literature [10] discusses the relationship between unsupervised and supervised machine learning, proposing a topic model based on open-source data that outperforms popular commercial

applications. Gender prediction algorithms are used to reveal clear topic differences between male and female scholars. However, this model primarily focuses on topic analysis for gender prediction, which may not be suitable for scenarios where the main data source is user behavior rather than text topics. Additionally, it may not fully exploit the sequential and contextual information in user behavior data.

In deep-learning technology, speech features are automatically generated through reinforcement learning of raw data, offering stronger recognition ability than manually generated features [11]. Entropy-based information theory and rough set theory (RST) are used to extract and select informative and accurate acoustic features related to gender recognition. A deep neural network model, composed of a CNN and a gated recurrent unit network (GRUN), is employed to extract useful features for gender recognition from audio-speech signals. Based on this feature vector, the hybrid gender-recognition model can effectively identify gender from speech signals. However, this approach primarily focuses on audio-speech data and cannot process other data types such as text or general user behavior. Furthermore, the model's performance may be influenced by the quality and diversity of the audio data.

Digital communication offers users the ability to select virtual gender through physical anonymity and improve author characterization by exploring the additional duality and biological-gender information in text-based document gender prediction. This method also compares and evaluates gender-prediction performance under various conditions and quantitatively assesses users' virtual and biological genders in real-time text-based online-messaging services [12]. The prediction accuracy reached 85.4%. However, this method is primarily suited for text-based online messaging scenarios and may not handle more complex user-behavior data from multiple sources.

This paper aims to explore a machine-learning-based approach to predict a user's gender by analyzing user behavior and other relevant characteristics. We propose a hybrid parallel deep-learning model based on CNN-BiLSTM-attention. Unlike previous studies, which often focus on a single data type (such as e-commerce, telecom, face-image, text, or audio data), this model can comprehensively process various types of user-behavior-related data. It combines the strengths of CNN for feature extraction, BiLSTM for capturing sequential dependencies, and an attention mechanism for emphasizing important features. This hybrid approach enables the model to better capture complex patterns in behavioral data, particularly temporal and contextual information. While previous models may struggle to handle sequential or contextual data, or fail to generalize across different data types, the multi-layer architecture of this proposed model outperforms simpler models in terms of prediction accuracy. It is adaptable to a wider range of data sources and user-behavior analysis scenarios, offering a more comprehensive and accurate solution for gender prediction.

3. Description of user consumption behavior characteristic data

The dataset used in this paper is sourced from the Tencent 2020 Advertising Algorithm Competition Dataset. The original dataset comprises 2.7 million training samples and 670,000 test samples, with identical data structures for both sets. Each sample represents a series of behaviors and related information generated by a user. The specific meanings of the features in the dataset are detailed in Table 1. The training and test datasets are stored in separate CSV format files. Given that the dataset contains missing values, imputation is employed to handle these gaps. The method used for imputation is mean imputation, where the missing values in each column are replaced with the

mean value of that column.

Table 1. The meaning of each feature in the data set.

Feature	Feature declaration
time	Day, granularity of time, integer value, value range [1,91]
userId	*userId* with random numbers from 1 to *N*, where *N* is the total number of users
creativeID	The ID of the AD material that the user clicked on
clickTime	The number of times the user clicked on the AD material that day
adID	The ID of the AD to which the material.
productID	The ID of the product advertised in the advertisement
productCategory	The category ID of the product advertised in this advertisement
advertiserID	ID of the advertiser
industry	The ID of the industry the advertiser belongs to
gender	User gender, value range [1,2]

After processing the missing values, user gender information can be analyzed and mined from the user's historical click data. In the training dataset, the user's advertisement (AD) click records are arranged chronologically to generate the user click stream data. Since the information represented by different IDs in the training set varies, the click streams for these IDs are generated separately to extract additional features. The click stream information for users is captured in chronological order, as illustrated in Table 2.

Table 2. Click flow information for users.

userID	clickFlow
1	821396 209778 877468 1683713 122032 71691 1940159 90171 2087846 ...
2	63441 155822 39714 609050 13069 441462 1266180 1657530 1696925 ...
3	661347 808612 710859 825434 593522 726940 392052 1173863 862241 ...
4	39588 589886 574787 1892854 1962706 2264105 1230094 31070 2348342 ...
5	296145 350759 24333 43235 852327 1054434 1054434 1296456 1248711 ...
	.....

Figure 1 shows the distribution of click stream data lengths. As can be seen from Figure 1, the lengths of the click streams vary, and a padding operation is required to either truncate or fill with zeros.

4. Materials and methods

4.1. Feature selection using ANOVA

Analysis of variance (ANOVA) is a statistical method used to test the differences in mean values between different groups. It evaluates the influence of independent variables on dependent variables by decomposing the variance in the data.

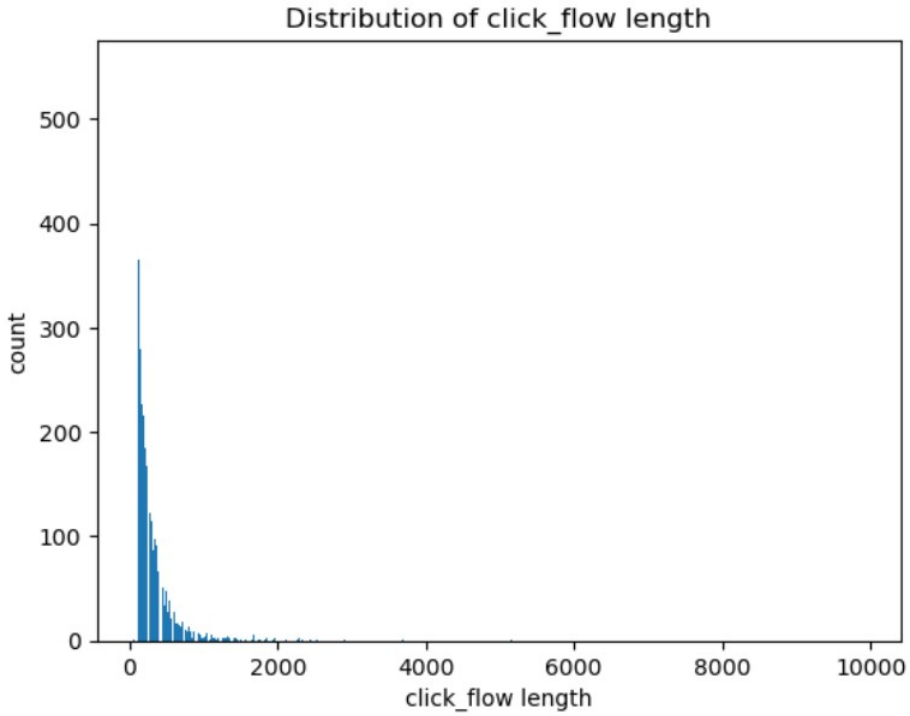

Figure 1. Distribution of click stream data length information.

4.1.1. The basic mode of ANOVA

The ANOVA model can be expressed as follows:

$$Y_{ij} = \mu + \tau_i + \epsilon_{ij} \tag{4.1}$$

where Y_{ij} represents the j-th observation in the i-th group, μ is the overall mean, and τ_i represents the treatment effect of the i-th group. ϵ_{ij} is the random error , assumed to follow $N(0, \sigma^2)$.

4.1.2. Hypothesis testing

Null hypothesis (H_0): All group means are equal, i.e., $\tau_1 = \tau_2 = \ldots \tau_k = 0$. Alternative hypothesis (H_1): At least one group mean is different.

4.1.3. Variance decomposition

The total variance (SST) is decomposed into between-group variance (SSB) and within-group variance (SSW):

$$SST = SSB + SSW \tag{4.2}$$

$$SST = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y})^2 \tag{4.3}$$

$$SSB = \sum_{i=1}^{k} n_i \left(\bar{Y}_i - \bar{Y}\right)^2 \tag{4.4}$$

$$SSW = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_i)^2 \tag{4.5}$$

4.1.4. F-statistic

The F-statistic is calculated to test the significance of group differences:

$$F = \frac{MSB}{MSW} \tag{4.6}$$

where MSB is the mean square between groups, MSW is the mean square within groups.

$$MSB = \frac{SSB}{k-1} \tag{4.7}$$

$$MSW = \frac{SSW}{N-k} \tag{4.8}$$

4.1.5. Decision rule

If $F > F_{\alpha,k-1,N-k}$, reject H_0, and otherwise, fail to reject H_0. If H_0 is rejected , a post-hoc test can be conducted to identify which specific groups differ.

The correlation between the features of this dataset is shown in Figure 2 and Table 3 below. A higher F-statistic indicates stronger evidence that the feature is related to the target.

Table 3. F-values and p-values for feature selection.

Feature	F-value	P-value
creativeID	297.721574	1.041753×10^{-66}
clickTime	6.493321	1.082811×10^{-02}
AdID	324.147680	1.826937×10^{-72}
ProductID	16025.218507	0
ProductCategory	3890.076634	0
AdvertiserID	1677.802357	0
industry	12758.699942	0

As can be seen from Table 3, the F-value of the creativeId feature is 297.72, and the p-value is very small (approximately 1.04×10^{-66}), indicating a significant linear relationship between the feature and gender. The F-value of the clickTimes feature is 6.49, and the p-value is about 0.01. Although the F-value is small, the p-value is still below the significance level of 0.05, suggesting that the linear relationship between the clickTimes feature and gender is statistically significant. The F-value of the adId feature is 324.15, and the p-value is very small (approximately 1.83×10^{-72}), indicating a significant linear relationship between the feature and gender. For productId, productCategory, advertiserId, and industry, the F-values of these features are very high, and the p-values are close to zero, indicating that the linear relationships between these features and gender are highly significant.

In summary, based on the results of ANOVA, it can be concluded that there is a significant linear relationship between these features and gender. In particular, productId, productCategory, advertiserId, and industry have a significant impact on gender.

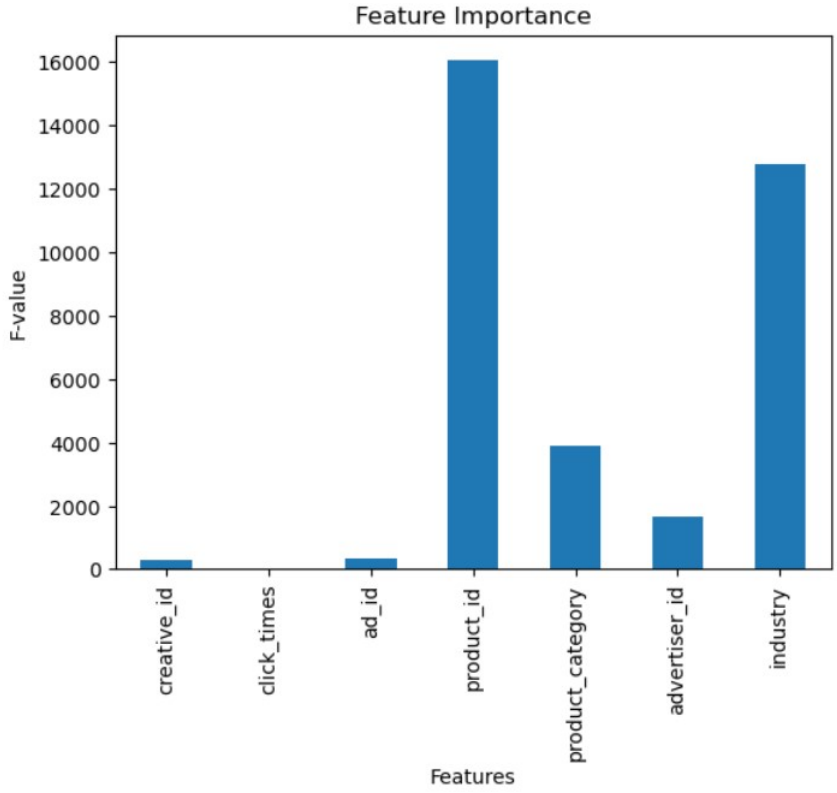

Figure 2. ANOVA F-number for each feature.

4.2. Feature selection using random forest models

Tree-based models [13] are more suitable than ANOVA for analyzing advertising click-stream data. First, click-stream data often exhibits non-linear characteristics, and tree-based models are well-equipped to handle non-linear relationships and complex interactions, whereas ANOVA assumes linearity. Second, tree-based models can automatically perform feature selection and rank feature importance, which helps identify key factors influencing clicks. In contrast, ANOVA mainly tests group differences and requires additional methods for feature selection. Lastly, tree-based models are more robust to noise and outliers in the data and can handle categorical and mixed-type data directly, which ANOVA cannot. For these reasons, feature selection in this paper is based on tree-based models.

4.2.1. Construction of decision trees

In a random forest, the construction of each decision tree is typically based on criteria such as information gain for classification trees [14]. For a dataset D and a feature A, the information gain *IG(D,A)* is defined as:

$$IG(D, A) = H(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} H(D_v) \tag{4.9}$$

where $(H(D) = -\sum_{i=1}^{n} p(C_i) \log_2 p(c_i)$ is the information entropy of the dataset D, $p(c_i)$ is the probability of class c_i in the dataset D, n is the number of classes, $(\text{values}(A))$ is the set of values of

feature A, D_v is the subset of D where feature A takes the value v, $\frac{|D_v|}{|D|}$ is the proportion of subset D_v in dataset D, and $H(D_v)$ is the information entropy of subset D_v.

Information gain measures the reduction in informational uncertainty after using feature A to partition the dataset D. The larger the information gain, the greater the contribution of this feature to classification, making it more suitable as a splitting node for the decision tree.

4.2.2. Construction of random forest

A random forest constructs multiple decision trees through bootstrap sampling and random feature selection.

Randomly sample n samples with replacement from the original dataset D to form a new dataset D_b (where b represents the b-th bootstrap sampling), which is used to construct the b-th decision tree. This sampling method makes the training data of each decision tree slightly different, increasing the diversity of the model.

When constructing each node of each decision tree, instead of considering all features, randomly select m features ($m \ll p$, where p is the number of original features), and then select the optimal splitting feature from these m features. This can further increase the differences between decision trees.

4.2.3. Prediction of random forest

For a new sample x, the prediction results of B decision trees in the random forest are $f_1(x), f_2(x), \cdots, f_B(x)$, respectively. The final predicted class y is determined by majority voting:

$$y = \text{argmax}_c \sum_{b=1}^{B} I(f_b(x) = c) \tag{4.10}$$

where $I(\cdot)$ is the indicator function. When the condition in the parentheses is true, $I(\cdot) = 1$; otherwise, $I(\cdot) = 0$.

Each decision tree in the random forest classifies and predicts the sample x. The random forest aggregates the prediction results from all the decision trees and selects the class that appears most frequently as the final predicted class.

The feature importance results after performing feature selection using a random forest are shown in Figure 3 and Table 4 below. Random forest evaluates feature importance by measuring how much each feature contributes to the splitting of nodes in the decision tree structure, reflecting their impact on predicting the target variable.

Creative_id (0.320354) and ad_id (0.309683) have the highest importance, suggesting that these features play the most significant role in the model's predictions. They are likely to be crucial because they directly relate to specific advertisements and their characteristics, which can effectively differentiate ad performance.

Advertise_id (0.133829) and industry (0.120221) also show notable importance, indicating that the advertiser and industry information are relevant for predicting click-through rates or ad effectiveness.

Product_id (0.074904) and product_category (0.023294) have relatively lower importance, likely because the product itself has less impact on ad effectiveness.

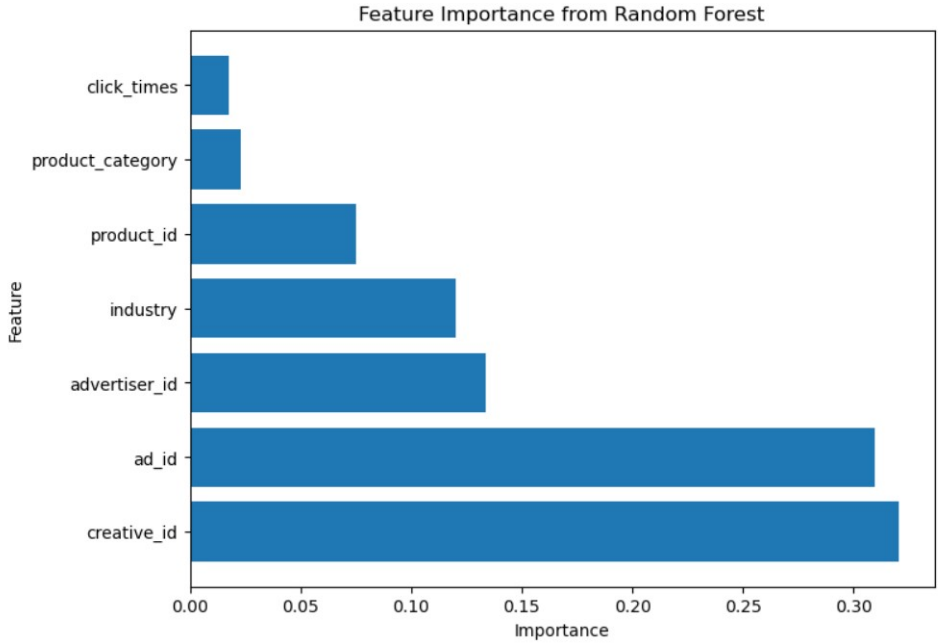

Figure 3. Bar chart of feature importance.

Table 4. Rank the importance of features.

0	creative_id	0.320354
2	ad_id	0.309683
5	advertiser_id	0.133829
6	industry	0.120221
3	product_id	0.074904
4	product_category	0.023294
1	click_times	0.017715

Click_times (0.017715) has the lowest importance, indicating that historical click counts have a weak influence on current ad performance, possibly because they lack context regarding the current ad placement.

These results help identify which features contribute most to the model's performance, providing valuable insights into feature selection and optimization for practical applications.

4.2.4. Data set unbalance verification

Verifying the imbalance in the dataset is crucial, as unbalanced data can lead to the model overfitting to the majority class, resulting in poor prediction performance on minority class samples. By checking for imbalance, potential issues can be identified early, allowing for corrective actions like oversampling, undersampling, or adjusting model weights. These steps can enhance the model's accuracy and generalization across various sample types, prevent bias, and enable the model to better reflect the true distribution and patterns in the data.

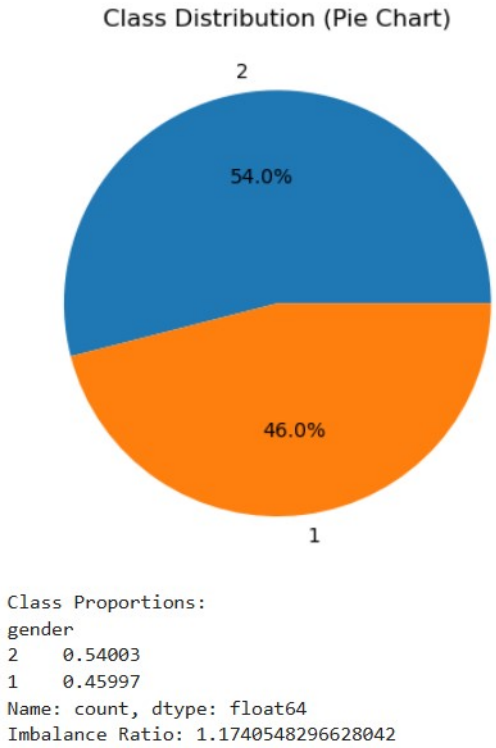

Figure 4. Bar chart of feature importance.

The imbalance verification results of the dataset are shown in the figure below. From the figure, it can be seen that the proportions of gender label 1 (male) and 2 (female) are 0.4599 and 0.54003, respectively, with an imbalance rate of 1.17. This indicates that the proportion difference between the two categories is small, suggesting that the dataset is relatively balanced across the categories. As a result, the model is less likely to encounter significant bias or overfitting issues during training, allowing it to treat all categories fairly and potentially achieve better generalization performance.

4.3. Word embedding technology

The clickstream data consists of discrete material IDs, which cannot be directly used for training. However, these data can be treated similarly to words and encoded using natural language processing techniques, specifically word embedding methods. By leveraging the context of each material ID, the clickstream data is transformed into word vectors. To avoid high dimensionality, these material IDs are represented by low-dimensional vectors. The similarity between different clickstream vectors is determined by calculating the cosine similarity between their corresponding word vectors, ensuring that vectors with similar meanings are closer together. This approach helps capture the semantic relationships within the clickstream data.

In word embedding methods, the continuous bag of words (CBOW) algorithm is employed to train the clickstream data. The CBOW model predicts the target word based on its surrounding context words. The fundamental concept is to use the vector representations of context words to estimate the probability distribution of the target word. Given a context window of size 2 m, the model processes

the context words to predict the target word's vector representation, and the vector representations of the context words are $W_{c-m}, W_{c-m+1}, \ldots, W_{c+m}$. The average on the context vectors is:

$$h = \frac{1}{2m} \sum_{\substack{i=-m \\ i \neq 0}}^{m} W_{c+i} \tag{4.11}$$

The probability of the target word W_t is computed using the softmax function:

$$P(W_t \mid W_{c-m}, \ldots, W_{c+m}) = \frac{\exp(W_t^T h)}{\sum_{k=1}^{V} \exp(W_k^T h)} \tag{4.12}$$

where W_t is the vector representation of the target word. V is the vocabulary size. The CBOW model aims to maximize the log-likelihood of the target word:

$$\tau = \sum_{t=1}^{T} \log P(W_t \mid W_{c-m}, \ldots, W_{c+m}) \tag{4.13}$$

where T is the number of training samples.

To reduce computational complexity, CBOW often uses negative sampling. The objective function is modified as:

$$\tau = \sum_{(t,c) \in D} \log \sigma(W_t^T h) + \sum_{(t,c) \in D'} \log \sigma(-W_t^T h) \tag{4.14}$$

where D is the positive sample set (target word and context word pairs), D' is the negative sample set (randomly sampled word pairs), and $\sigma(\cdot)$ is the sigmoid function.

The CBOW model predicts the target word based on context words, using the average of context vectors and the softmax function to compute the probability distribution of the target word. Negative sampling is often used to improve computational efficiency.

The steps of the word embedding algorithm are described as follows:

Step 1: Prepare the click stream sample without click records and convert the click stream data of each user from characters to a list of component words.

Step 2: Set the minimum word frequency, word vector length, and context window size, and then train the click stream list using the CBOW algorithm.

Step 3: Initialize a matrix to store all vectors, where the number of rows is the total number of words +1, the number of columns is the word vector dimension, and it holds the word vectors for all words.

Step 4: Encode and pad the words in the word vector matrix.

Step 5: Sort and store the word vector matrix accordingly.

5. Hybrid parallel architecture for gender recognition

The hybrid parallel architecture, as illustrated in Figure 5, consists of the embedding phase, the first parallel BiLSTM phase, the second parallel CNN phase, and the merging phase.

By integrating CNN, BiLSTM, and attention mechanisms, the model is able to extract features at multiple levels. The CNN captures local features through convolutional kernels of varying sizes, while

the BiLSTM provides a global context across time. This combination allows the model to capture both local patterns and global temporal dependencies, enhancing its ability to understand complex data.

The hybrid architecture takes advantage of the strengths of both the CNN and BiLSTM, making it more effective in many applications compared to using either model in isolation. This is especially beneficial when both spatial and temporal features are crucial. The multi-layer BiLSTM captures temporal dependencies, while the CNN excels at extracting local spatial features. The final concatenation step merges both feature types, resulting in improved overall performance.

Incorporating an attention layer into the architecture brings significant benefits. It allows the model to focus on the most important parts of the input sequence, improving its ability to capture key features. By applying the attention mechanism to the RNN output and performing pooling operations, the model's representational power is strengthened, enabling it to learn data patterns more accurately and, as a result, enhancing prediction accuracy.

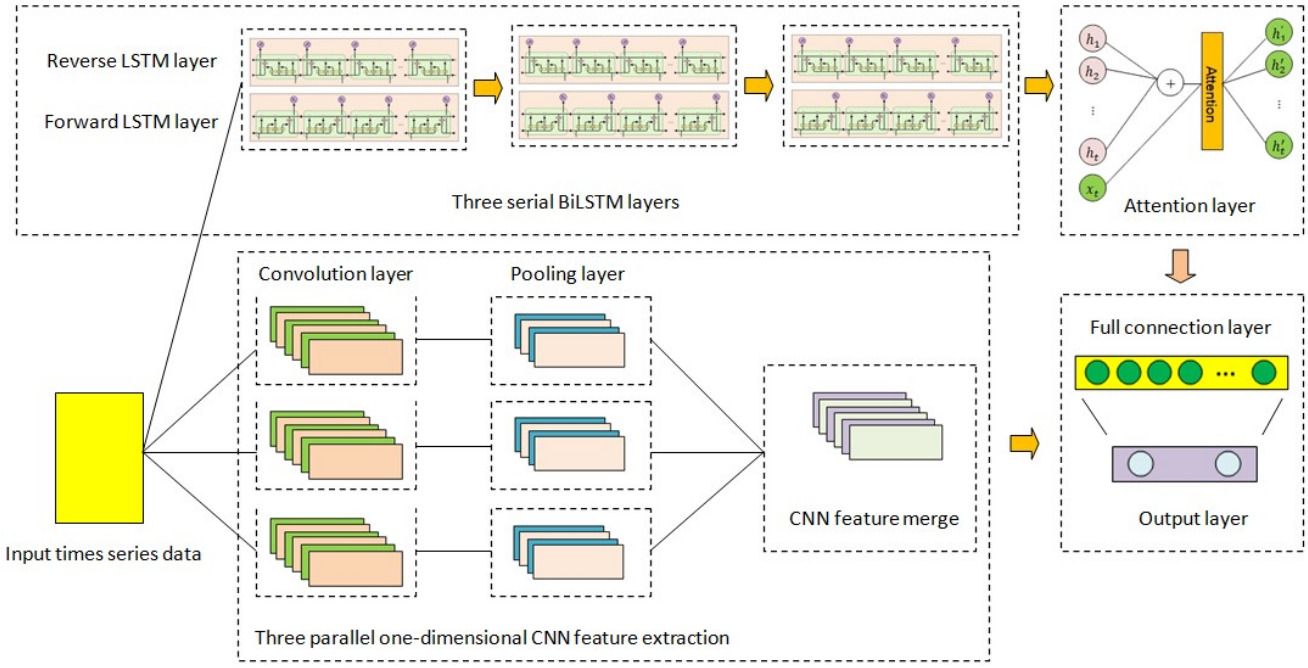

Figure 5. Hybrid parallel architecture.

5.1. CNN principle structure

The CNN is a type of deep neural network commonly used in image recognition [15–17], natural language processing [18], and speech recognition [19]. It extracts features and reduces the dimensionality of input data through convolution operations, enabling efficient processing of complex data. The 1D convolutional neural network (1D-CNN), a variant of CNN, is particularly effective in tasks such as speech recognition, natural language processing, and time series prediction, as it excels at capturing local relationships within sequence data.

This paper utilizes a parallel CNN component consisting of three one-dimensional CNNs to extract feature information. The convolution kernels used are of sizes 3, 4, and 5, which facilitate

feature extraction at different scales. A global maximum pooling layer is applied to the convolutional results, selecting the maximum value from each filter. These pooled values are then concatenated and combined with the results from the RNN.

5.2. BiLSTM model

LSTM is a variant of the RNN designed to address the issues of gradient vanishing and gradient explosion that occur when traditional RNNs process long sequence data. By incorporating gating mechanisms, LSTM can effectively capture and retain long-term dependencies.

The fundamental component of LSTM is the memory cell, which consists of the input gate, the forget gate, and the output gate. These gates control the flow of information, enabling the network to maintain long-term memory and perform selective forgetting. The corresponding calculation formulas for these components are given in Eqs (5.1)–(5.6):

$$i_t = \sigma\left(w_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{5.1}$$

$$\tilde{C}_t = \tanh\left(w_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{5.2}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{5.3}$$

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{5.4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{5.5}$$

$$h_t = o_t \odot \tanh(C_t) \tag{5.6}$$

where i_t, f_t, o_t denote the input gate, forgetting gate, and output gate, respectively. w_i, w_C, W_f, W_o represent weight matrices. b_i ,b_C, b_f, b_o are bias terms. σ is the sigmoid activation function. C_t is the candidate cell state vector, and C_t is the updated cell state.

LSTM can process information in only one direction of the data sequence, while the BiLSTM [20, 21] combines both forward and backward LSTMs. This configuration places one LSTM before and after each training sequence, with the two unidirectional LSTMs connected at the same layer. The network can encode not only the previous information at the current time step but also the subsequent information, capturing the relationships of feature changes throughout the sequence. This allows the model to predict the next state of the sequence more accurately by considering both forward and backward directions. The BiLSTM model structure [22] is shown in Figure 6.

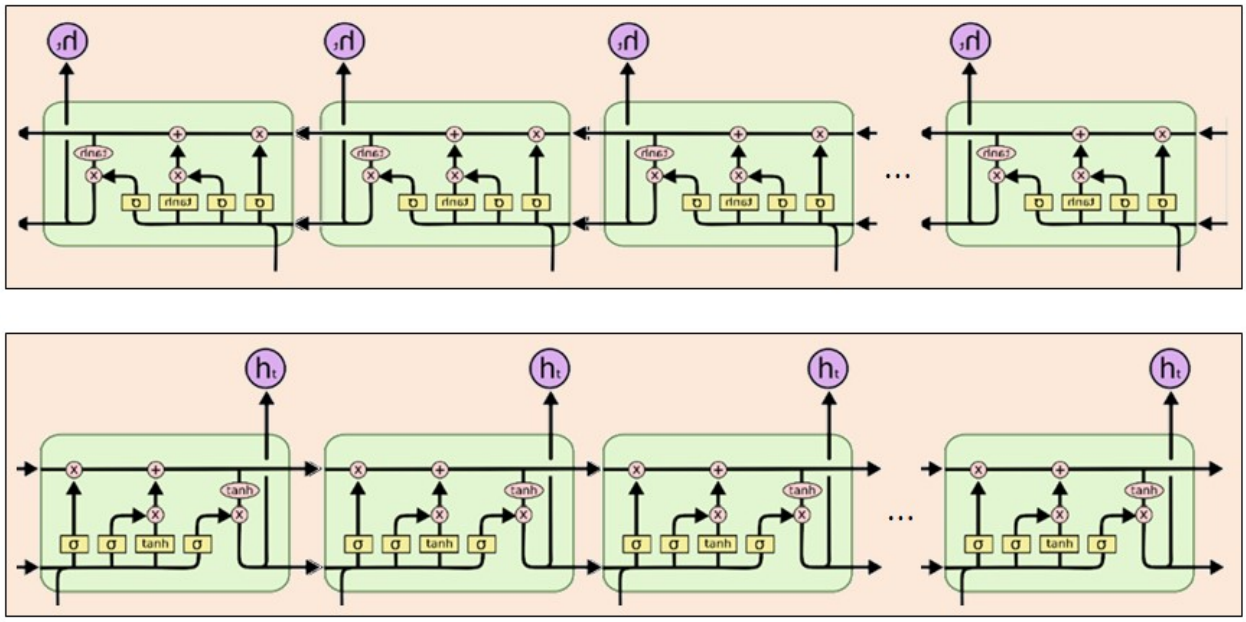

Figure 6. BiLSTM network structure.

5.3. Attention mechanism

In the attention mechanism, each element of the input sequence is assigned a unique weight, which is determined by calculating the correlation between different parts of the sequence. When the context information provided by BiLSTM is combined with the attention mechanism, it significantly enhances the model's performance across various tasks, particularly those that require capturing long-term dependencies. This combination improves both the accuracy and quality of the model's outputs. The structure of the attention mechanism unit is shown in Figure 7.

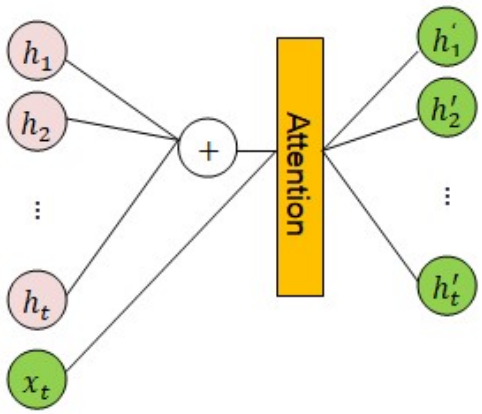

Figure 7. Attention mechanism structure [23].

The attention mechanism is calculated as follows:

(1) Calculate the similarity scores.

For the scaled dot-product attention, first calculate the dot product between the Query and Key. Here, the Query, Key, and Value are all X, and the similarity score matrix $S \in \mathbb{R}^{n\times n}$ is obtained as $S = XX^T$.

(2) Scale the scores.

Since use_scale = True, the scores need to be scaled. The scaling factor is $\sqrt{d}$, and the scaled score matrix is $S_{scaled} = \frac{S}{\sqrt{d}}$.

(3) Apply the Softmax function.

Apply the Softmax function to the scaled score matrix to convert the scores into a probability distribution, obtaining the attention weight matrix $A \in \mathbb{R}^{n\times n}$. The formula is $A = \text{Softmax}(S_{scaled})$ where the definition of the Softmax function is $\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$.

(4) Calculate the attention output.

Multiply the attention weight matrix by the value matrix (here the value matrix is also X) to obtain the attention output matrix $O \in \mathbb{R}^{n\times d}$. The formula is: $O = AX$.

(5) Global average pooling operation.

After obtaining the attention output $O \in \mathbb{R}^{n\times d}$ through the above steps, perform the global average pooling operation. Global average pooling calculates the average of each feature dimension over the sequence length dimension, resulting in a vector O_{pooled} with a dimension of $\mathbb{R}^d$. The formula is: $O_{pooled} = \frac{1}{n}\sum_{i=1}^{n} O_i$ where O_i is the i-th row of the attention output matrix O.

The entire attention calculation process (including attention calculation and global average pooling) can be expressed as:

$$O_{pooled} = \frac{1}{n}\sum_{i=1}^{n}\left(\text{Softmax}\left(\frac{XX^T}{\sqrt{d}}\right)X\right)_i \tag{5.7}$$

5.4. Prediction process of the CNN-BiLSTM-attention parallel model

The model first uses an embedding layer to map a sequence of click streams into a 100×64 matrix vector, which is then fed as input to the first BiLSTM layer and the parallel Conv1D layer.

Through the operation of the three-layer BiLSTM network, the matrix is transformed into a 60-dimensional feature vector, and a 60-dimensional context vector is generated using the attention mechanism.

Three parallel convolutions, followed by a maximum pooling operation, filter out the important local features. The outputs from the three parallel convolutions are concatenated into a 30-dimensional feature vector. The output from the attention mechanism is then merged with the results of the parallel convolution operations, and the combined output forms a 90-dimensional feature vector, which is activated by the Softmax function.

Once the characteristic information is obtained, a specific value is generated, corresponding to the predicted result. The BP algorithm is then used to compute the loss between the actual and predicted values, and the parameters are updated layer by layer for the next round of training.

6. Experiment and analysis

The experimental environment in this paper uses the Windows 11 system, Python 3.11 as the programing language, and the Keras framework based on TensorFlow 2.12.0. The detailed

environment configuration is shown in Table 5.

Table 5. Experimental environment configuration table.

Name	Specification and model
Operating system	Windows11
Programing environment	Python3.11
Deep learning framework	Tensorflow2.12.0
Natural language processing library	Gensim
Numerical calculation	Numpy1.23.5

The parameters for Word2Vec training are as follows: The minimum word frequency is set to 2, meaning that words with a frequency lower than 2 will be filtered out and will not participate in the training process. This helps minimize the impact of low-frequency words on the training results.

The context window size is set to 3, indicating that during the word vector training, the maximum distance between context words and the current word is 3 words. Specifically, for a given center word, the 3 words before and after it are used as context to train the word vectors, which helps capture the local semantic relationships between words.

The word vector dimension is set to 64. This parameter defines the number of dimensions for each word in the vector space. A higher dimension allows the word vector to represent more information, but it also increases computational complexity and storage requirements.

The training algorithm (sg) is set to 0, which indicates the use of the CBOW (continuous bag-of-words) algorithm for word vector training. Additionally, the skip-gram method is tested and verified. Experimental results show that whether the sg parameter is set to 0 or 1, it has a negligible impact on the efficiency of word vector training.

In the CNN section of the model, the kernel sizes are set to (3, 4, 5), and the number of channels is set to 10. The ReLU activation function is chosen, as it effectively mitigates the vanishing gradient problem and enhances the network's ability to learn non-linear representations. The global max-pooling method is used to extract the maximum value from the feature map, retain the most prominent features, and reduce the feature dimensions.

In the BiLSTM section of the model, the number of bidirectional LSTM units in each layer is set to 100, 60, and 30, respectively. This decreasing configuration helps refine features progressively while reducing the number of parameters and computational complexity. The bidirectional structure enables the model to learn information from both the preceding and succeeding parts of the sequence, improving its ability to understand the context.

For gender prediction, the Adam optimizer and sparse categorical cross-entropy loss function are used to evaluate accuracy. The sparse categorical cross-entropy loss function is appropriate when the target labels are integers. Instead of using one-hot encoding, sparse categorical cross-entropy directly compares the predicted probability distribution to the integer class label.

In gender prediction, the model learns to predict the correct gender by minimizing the loss, which represents the negative log-likelihood of the correct class. The lower the loss, the more accurate the model's predictions are. Sparse categorical cross-entropy is particularly suitable for binary classification tasks where each input has a single class label.

During model training, the model.fit method is called with the clickstream and gender label data, after word vector training, as input. The model is trained for 16 epochs, processing 16 samples per batch, with 10% of the data used as a validation set. To monitor the model's performance on the validation set, the learning rate is set to 0.001, which is the default learning rate for the Adam optimizer.

6.1. Evaluation criteria

The proposed model uses precision and recall [24, 25] as metrics to evaluate the results of gender classification for positive (correct gender) and negative (incorrect gender) samples, respectively. Accuracy is also used as a fundamental metric, representing the proportion of correct predictions among all predictions. True positives (TPs) refer to correctly predicted positive instances, true negatives (TNs) represent correctly predicted negative instances, false positives (FPs) refer to incorrectly predicted positive instances, and false negatives (FNs) refer to incorrectly predicted negative instances.

Precision measures the proportion of true positive predictions out of all instances predicted as positive. It is calculated as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6.1}$$

Recall measures the proportion of true positive predictions out of all the actual positive instances, and is calculated as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6.2}$$

Accuracy measures the proportion of correct predictions (both true positives and true negatives) out of all predictions, and is calculated as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.3}$$

F1 score is the harmonic mean of precision and recall, providing a balanced measure when there is an uneven class distribution. It is calculated as follows:

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{6.4}$$

6.2. Effect of the epoch parameter on model results

In machine learning and deep learning, an epoch refers to one complete pass of the entire training dataset through the model. During each epoch, the model learns from the training data by performing both forward and backward passes, adjusting its parameters to minimize the loss function. With each epoch, the model gradually identifies patterns in the data, improving its ability to make predictions on new, unseen data. However, using too many epochs can lead to overfitting, where the model memorizes the training data and performs poorly on new data. Conversely, too few epochs may result in underfitting, where the model fails to learn enough from the data. The accuracy over the epochs is shown in Figure 8.

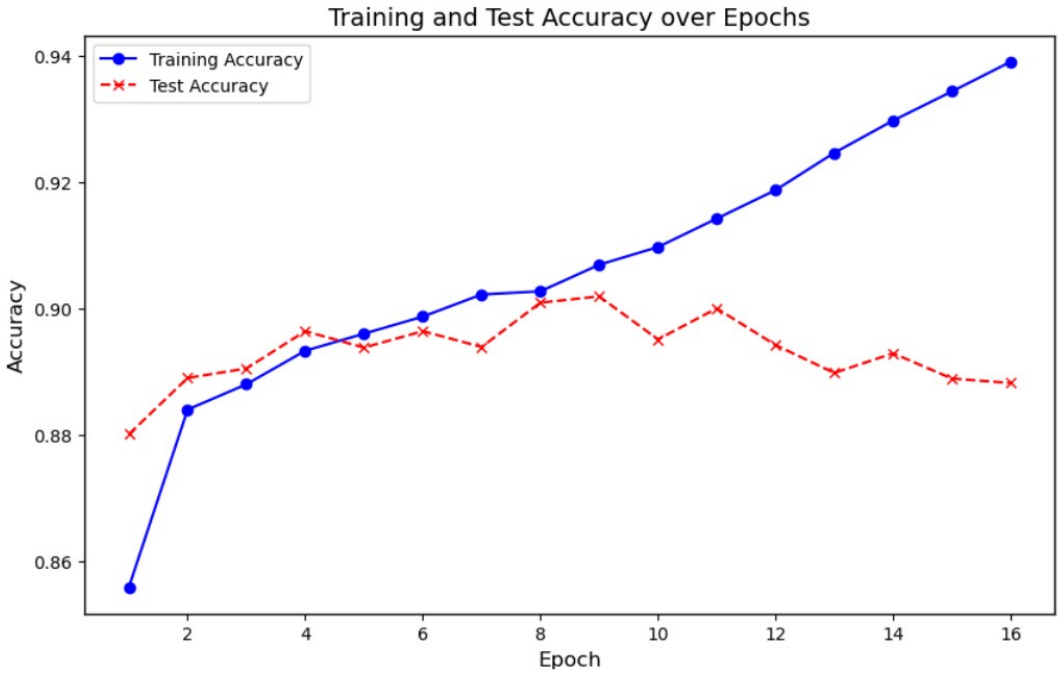

Figure 8. Comparison of training results under different epochs.

As shown in Figure 8, the training set accuracy steadily improves with each epoch, while the test set accuracy initially increases and then begins to decline. The peak test set accuracy occurs at the 7th epoch, after which it starts to decrease, potentially indicating overfitting.

6.3. Comparison of training results of the CNN, RNN, LR, transformer, and our model

The performance of the model was validated using the 5-fold cross-validation method. This technique divides the training dataset into five equal parts, with four parts used for training and one part reserved for testing. The final evaluation metric is the average of the accuracies from the five folds. Accuracy (acc), precision, recall, and F1-score were used to assess the model's performance.

To verify the effectiveness and performance advantages of the proposed method, the dataset used in this paper was applied. The methods compared include the CNN, RNN, LR, and the transformer model [26] for classification prediction. A comparative experiment with a 50% crossover test was conducted, and the experimental results are presented in Table 6.

Table 6. Comparison of classification results based on different methods.

Model	ACC	P	R	F1
Ours	0.948	0.932	0.911	0.921
CNN	0.887	0.858	0.803	0.826
RNN	0.930	0.908	0.877	0.892
LR	0.818	0.792	0.607	0.687
Transformer	0.878	0.886	0.879	0.880

The results presented in Table 6 clearly demonstrate the superior performance of the proposed method compared to other individual models across four key indicators: accuracy, precision, recall, and F1 score. Specifically, its accuracy exceeds that of the CNN model by 6.1%, the RNN model by 1.8%, the LR model by 13.0%, and the transformer model by 7.0%. These results strongly suggest that the proposed method excels in gender prediction tasks.

Among the models evaluated, the LR model exhibits relatively weak performance, with an accuracy of 0.818, precision of 0.792, recall of 0.607, and an F1 score of 0.687. The low values across these indicators indicate that the LR model struggles with classification, particularly in identifying positive samples, which leads to a low recall rate and consequently impacts its precision.

The transformer model demonstrates moderate performance, achieving an accuracy of 0.878, precision of 0.886, recall of 0.879, and an F1 score of 0.880. While the transformer model performs well in terms of precision, it still falls short of the proposed method in terms of both accuracy and recall.

Our fusion model, which combines the strengths of the CNN and BiLSTM, effectively extracts features and addresses challenges such as gradient explosion and vanishing gradients. More importantly, it excels at capturing long-range dependencies, allowing it to outperform other individual models in all key indicators, making it particularly well-suited for complex gender prediction tasks.

6.4. Statistical significance testing in performance between models

Statistical significance testing is an important part of model evaluation. In the comparison between the CNN model and Our Model, the RNN model and Our Model, the logistic regression model and Our Model, and the transformer model and Our Model, the t-test was performed to assess the differences in the false positive rate (FPR) and true positive rate (TPR). The results of the statistical tests are shown in Table 7.

Table 7. Statistical test results of FPR and TPR differences between Our Model and Other Models.

Comparsion model	t-value of FPR	p-value of FPR	t-value of TPR	p-value of TPR
CNN model vs. Our model	−8.562	2.1×10^{-14}	−9.377	2.503×10^{-15}
RNN model vs. Our model	−3.698	3.57×10^{-4}	−3.698	3.5684×10^{-4}
LR model vs. Our model	25.973	5.415×10^{-46}	25.974	6.415×10^{-46}
Transformer model vs. Our model	−7.582	2.84×10^{-4}	−7.583	4.7384×10^{-4}

6.5. Key considerations for model deployment

The experimental results of the model showcase several key performance metrics that are crucial for evaluating its suitability for practical applications. The average training time is 25.65 seconds, reflecting the time spent by the model during the training phase. Training time is influenced by factors such as model architecture complexity, the size of the training dataset, and the computing resources available. A shorter training time allows for faster model iterations during development, thereby enhancing R&D efficiency.

The average inference time is 0.46 seconds, representing the time required for the model to make predictions on new input data. In application scenarios with high real-time requirements, such as

real-time monitoring systems or online recommendation engines, a fast inference time ensures timely responses and provides real-time services to users. The model's average inference speed is 43,778.26 inferences per second, further emphasizing its ability to efficiently process new data and perform a large volume of inference tasks per unit of time.

Furthermore, the average training memory usage is 4.92 MB, while the average inference memory usage is 0.69 MB. Memory consumption is a key consideration when deploying the model. The model's low memory requirements enable it to run on devices with limited resources, such as edge computing devices or servers with constrained memory, thereby expanding its application scope and reducing deployment costs and complexity.

6.6. ROC curve

The ROC curve is created by plotting the TPR on the y-axis and the FPR on the x-axis across various classification thresholds. This curve provides insight into the trade-off between sensitivity and specificity at different threshold values. A model with a curve closer to the top-left corner indicates better performance, while a random classifier would produce a diagonal line from (0, 0) to (1, 1). The area under the ROC curve (AUC) represents the overall performance of the classifier, with a value closer to 1 indicating superior performance. Figure 9 displays the ROC curves for various methods applied to the test set.

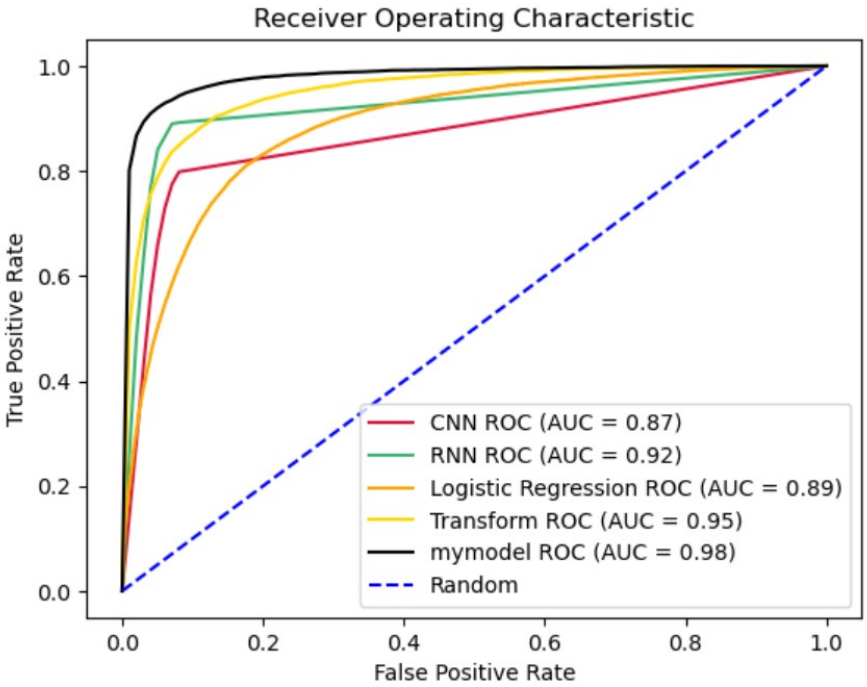

Figure 9. ROC curves of different methods on the test set.

As shown in the figure, the proposed model outperforms the individual machine learning models, achieving the largest area under the ROC curve, with an AUC value of 0.98. This indicates that the model's classification performance is highly significant.

6.7. Ablation study

To assess the contribution of the individual CNN, BiLSTM, and attention mechanisms to the final prediction accuracy of the model, an ablation experiment is conducted. The results showing the contribution of each mechanism to the model's prediction accuracy are presented in Table 8 below.

Table 8. Ablation study.

Model	ACC	P	R	F1
Our Model	0.948	0.932	0.911	0.921
onlyCNN	0.942	0.916	0.910	0.913
onlyBiLSTM	0.923	0.883	0.887	0.884
onlyAttention	0.813	0.739	0.681	0.708

Based on the data above, the contribution rates of the individual models—onlyCNN, onlyBiLSTM, and onlyAttention—can be calculated. Assume that the accuracy (ACC) of Our Model, which combines these three models, is 0.95. To calculate the contribution rate of each model (onlyCNN, onlyBiLSTM, and onlyAttention) to Our Model, we use the following method: the contribution rate is determined as the proportion of the ACC of each individual model relative to the performance of Our Model. The calculation formula is as follows:

$$Contribution\ rate = \frac{Performance\ metrics\ for\ individual\ models}{Sum\ of\ all\ model\ performance\ indicators} \times 100\% \tag{6.5}$$

Based on the results of the ablation study, the contribution rates of onlyCNN, onlyBiLSTM, and onlyAttention to the hybrid model Our Model are 35.1%, 34.4%, and 30.5%, respectively. Among these, the CNN model has the highest contribution rate, followed by BiLSTM, while the attention mechanism has the lowest contribution rate. This suggests that the CNN and BiLSTM play more significant roles in the hybrid model than attention.

7. The idea of model deployment to a real-world recommendation system

Deploying the CNN-BiLSTM-attention model (Our Model) in real-world recommendation systems involves several key steps.

(1) Model optimization and packaging: Since the trained model may be large in size, techniques such as quantization and pruning are necessary to reduce the number of parameters and computational complexity, thus minimizing the model's size. The optimized model is then packaged into a format suitable for the target deployment environment, such as the SavedModel format supported by TensorFlow Serving, which facilitates subsequent loading and invocation.

(2) Selection of the deployment architecture: If the recommendation system needs to handle high-concurrency requests, distributed deployment can be adopted. A load balancer is used to evenly distribute user requests across multiple servers running the model, improving the system's overall response speed. In scenarios with extremely high real-time requirements, such as online advertising recommendations, the model can be deployed at edge nodes close to the data source to reduce data transmission latency.

(3) Model integration and monitoring: Integrate Our Model into the overall architecture of the recommendation system, ensuring it works in coordination with data processing modules, user profiling modules, and others. Meanwhile, establish a comprehensive monitoring system to track indicators such as the model's inference time, accuracy, and memory usage in real-time. If model performance deteriorates or anomalies occur, timely adjustments or retraining can be made to ensure the stable and efficient operation of the recommendation system.

8. Conclusions and future work

With the rapid advancement of deep learning technology, gender prediction has shifted from traditional manual feature extraction methods to automatic models driven by deep learning. Gender prediction is now widely applied across various domains, including computer vision, natural language processing, social behavior analysis, and virtual assistants. To enhance the accuracy and robustness of gender prediction, this paper proposes a hybrid deep learning architecture that combines three parallel CNNs, three serial BiLSTMs, and an attention mechanism. The core design of the system leverages CNNs for extracting local features, BiLSTMs to model contextual relationships in sequential data, and the attention mechanism to assign higher weights to important features. Finally, gender classification is performed by concatenating the outputs of these two feature types.

Future research could explore the following directions for improvement:

(1) Improving computational efficiency: The model can be compressed through pruning and quantization, reducing both its size and inference time, thereby enhancing computational efficiency.

(2) Incorporating multimodal data: The model could be further optimized by integrating multimodal data, such as images, text, and speech, to improve its robustness and prediction accuracy.

(3) Leveraging self-supervised learning: In situations where data annotation is challenging, self-supervised learning techniques can be explored to reduce reliance on annotated data and enhance the model's ability to generalize.

(4) Visualizing attention weights: Attention weights can be visualized using techniques like heatmaps or attention maps, providing insights into which parts of the input the model focuses on during processing.

(5) Applying explainable AI (XAI): Future work could investigate the use of XAI techniques to analyze the reasoning behind the model's specific predictions.

(6) Comparing with lightweight models: The performance of the current model can be compared with that of lightweight models designed for mobile or edge computing, assessing efficiency and scalability.

(7) Expanding the dataset: In future studies, the dataset could be expanded to include additional factors such as age, geographic location, and device type. This would enable a more comprehensive analysis and evaluation, offering a deeper understanding of potential biases and improving the model's ability to generalize across diverse populations.

Author contributions

Lu Xiaoping: provide theoretical guidance and data analysis support; review the thesis and put forward revision suggestions. Wang Zichang: conceptualization, methodology, Data analysis,

writing–original draft and other works.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The work was supported by the Guangdong Natural Science Foundation (Grant #2023A1515012044).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. W. Nadeem, D. Andreini, J. Salo, T. Laukkanen, Engaging consumers online through websites and social media: A gender study of Italian generation Y clothing consumers, *Int. J. Inf. Manage.*, **35** (2015), 16–64. http://doi.org/10.1016/j.ijinfomgt.2015.04.008
2. B. Hasan, Exploring gender differences in online shopping attitude, *Comput. Hum. Behav.*, **26** (2010), 33–74. http://doi.org/10.1016/j.chb.2009.12.012 ,596–601.
3. M.Zhou, Gender difference in web search perceptions and behavior: Does it vary by task performance?, *Comput. Educ.*, **78** (2014), 174–184.http://doi.org/10.1016/j.compedu.2014.06.005
4. C. McLaughlin, L. Bradley, G. Prentice, E. J. Verner, S. Loane Gender differences using online auctions within a generation Y sample: An application of the theory of planned behaviour, *J. Retailing Consum. Services.*, **56** (2020), 1–13. http://doi.org/10.1016/j.jretconser.2020.102181
5. Z. Huang, J. Mou, Gender differences in user perception of usability and performance of online travel agency websites, *Technol. Soc.*, **66** (2021).http://doi.org/10.1016/j.techsoc.2021.101671
6. S. Choudhary, M. Agarwal, Framework for gender recognition using facial features by using deep learning, in *Second International Conference on Image Processing and Capsule Networks: ICIPCN 2021*, **300** (2022), 599–608. https://doi.org/10.1007/978-3-030-84760-9_51
7. T. Van Hamme, G. Garofalo, E. Argones Rúa, D. Preuveneers, W. Joosen, A systematic comparison of age and gender prediction on imu sensor-based gait traces, *Sensors*, **19** (2019), 2945–2961. https://doi.org/10.3390/s19132945
8. S. Haseena, S. Saroja, R. Madavan, A. Karthick, B. Pant, M. Kifetew, Prediction of the age and gender based on human face images based on deep learning algorithm, *Comput. Math. Methods Med.*, **2022** (2022). https://doi.org/10.1155/2022/1413597
9. M. M. Islam, J. H. Baek, Deep learning based real age and gender estimation from unconstrained face image towards smart store customer relationship management, *Appl. Sci.*, **11** (2021), 4549. http://doi.org/10.3390/app11104549

10. K. ELKarazle, V. Raman, P. Then, Facial gender classification using deep learning, in *Proceedings of the 13th International Conference on Soft Computing and Pattern Recognition.*, **417** (2022),598–607. http://doi.org/10.1007/978-3-030-96302-6 .

11. G. Yasmin, A. K. Das, J. Nayak, S. Vimal, S. Dutta, A rough set theory and deep learning-based predictive system for gender recognition using audio speech, *Soft Comput.*, **28** (2022),1–24. http://doi.org/10.1007/s00500-022-07074 .

12. P. C. S. Reddy, K. Sarma, A. Sharma, P. V. Rao, S. G. Rao, G. R. Sakthidharan, et al., Enhanced age prediction and gender classification (EAP-GC) framework using regression and SVM techniques, *Mater. Today Proc.*, **2020** (2020). http://doi.org/10.1016/j.matpr.2020.10.857

13. Y. Efea, L. Demirb, The impact of feature selection models on the accuracy of tree-based classification algorithms: Heart disease case, *Proc. Comput. Sci.*, **253** (2025), 757–764. https://doi.org/10.1016/j.procs.2025.01.1.

14. Z. Wang, Research on feature selection methods based on random forest, *Tehnički Vjesnik*, **30** (2023), 623–633.

15. M. A. Rahman, M. R. Islam, M. A. H. Rafath, S. Mhejabin, CNN based covid-19 detection from image processing, *J. ICT Res. Appl.*, **17** (2023). http://doi.org/10.5614/itbj.ict.res.appl.2023.17.1.7

16. A. Briot, P. Viswanath, S. Yogamani, Analysis of efficient cnn design techniques for semantic segmentation, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, **2018** (2018), 1065–1089. http://doi.org/10.1007/978-3-030-01234-2_1

17. A. Anton, N. F. Nissa, A. Janiati, N. Cahya, P Astuti, Application of deep learning using convolutional neural network (CNN) method for Womenâ€™ s skin classification, *Sci. J. Inf.*, **8** (2021), 144–153. http://doi.org/10.15294/sji.v8i1.26888

18. A. Kumar, R. Mamgai, R. Jain, Application of IoT-enabled CNN for natural language processing, in *IoT-enabled Convolutional Neural Networks: Techniques and Applications*, River Publishers, (2023), 149–177. https://doi.org/10.1201/9781003393030-6

19. A. Dutt, P. Gader, Wavelet multiresolution analysis based speech emotion recognition system using 1D CNN LSTM networks, in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **31** (2023), 1–12. http://doi.org/10.1109/TASLP.2023.3277291

20. J. Ramesh, R. Manavalan, Prostate ultrasound image classification using CNN-BILSTM, *Indian J. Comput. Sci. Eng.*, **12** (2021), 1611–1620. http://doi.org/10.21817/indjcse/2021/v12i6/211206028

21. D. H. Fudholi, R. A. N. Nayoan, A. F. Hidayatullah, D. B. Arianto, A hybrid cnn-BiLSTM model for drug named entity recognition, *J. Eng. Sci. Technol.*, **17** (2022), 730–744.

22. M. Yanga, J. Wang, Adaptability of financial time series prediction based on BiLSTM, *Proc. Comput. Sci.*, **199** (2022), 18–25. http://doi.org/10.1016/j.procs.2022.01.003

23. J. J. Ren, H. Wei, Z. L. Zou, T. T. Hou, Y. L. Yuan, J. Q. Shen, et al., Ultra-short-termpower load forecasting based on CNN-BiLSTM-attention, *Power Syst. Prot. Control*, **50** (2022), 108–116. https://doi.org/10.19783/j.cnki.pspc.211187

24. Q. H. Kha, T. O. Tran, V. N. Nguyen, K. Than, N. Q. K. Le, An interpretable deep learning model for classifying adaptor protein complexes from sequence information, *Methods*, **207** (2022), 90–96. https://doi: 10.1016/j.ymeth.2022.09.007

25. Q. H. Kha, V. H. Le, T. N. K. Hung, N. Q. K. Le, Development and validation of an efficient MRI radiomics signature for improving the predictive performance of 1p/19q co-deletion in lower-grade gliomas, *Cancers*, **21** (2021), 5398. https://doi.org/10.3390/cancers13215398

26. S. Madan, M. Lentzen, J. Brandt, D. Rueckert, M. Hofmann-Apitius, H. Fröhlich, Transformer models in biomedicine, *BMC Med. Inf. Decis. Making*, **24** (2024), 214. https://doi.org/10.1186/s12911-024-02600-5

AIMS Press