Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **9**, Number **2**, May **2024**

Research on Key Technologies of Short Video Based on Attention Mechanism and Deep Learning

Yan Lu*

School of Business
Guangzhou Vocational College of Science and Technology and Trade
Guangzhou 511442, P. R. China
76263534@qq.com

Dong-Jian Yang

School of Graduate Studies
Lingnan University, Hong Kong 999077, P. R. China
dongjianyang@ln.edu.hk

*Corresponding author: Yan Lu
Received September 25, 2023, revised December 9, 2023, accepted February 24, 2024.

Abstract. *With the popularity of the Internet and mobile devices, users generate a large amount of short video content every day, including social media platforms and online video platforms. This large amount of data requires intelligent and personalised recommendations to meet user needs. Therefore, a short video recommendation algorithm based on attention mechanism and neural network is proposed. Firstly, the attention mechanism is introduced on the basis of recurrent neural networks to adaptively learn the output weights of second-order features, which solves the problem that invalid second-order features may bring noise and adversely affect the model performance. Then, when calculating the attention, the second-order feature vectors and the target video vectors are equally sliced into a number of vectors, constituting a number of different hidden semantic spaces, so as to calculate the attention weights. The problem of weight smoothing between vector elements is improved by learning the relationship between feature vectors and target short video vectors at a fine-grained level. Experimental results show that the attention mechanism overcomes the output weighting problem of recurrent neural networks. Compared with the commonly used video recommendation algorithms, by reasonably setting the window size of the attention mechanism, the proposed algorithm exhibits higher recommendation performance under different feature vector size samples.*
Keywords: video recommendation; attention mechanism; recurrent neural network; output weights; adaptive

1. **Introduction.** With the popularity of the Internet and mobile devices, users generate a large amount of short video content every day, including social media platforms and online video platforms. This large amount of data requires intelligent and personalised recommendations to meet user needs [1, 2].

Different users have different preferences and interests in short videos. Artificial neural network technology can analyse users' interests, viewing history and interaction behaviours to construct user profiles [3, 4], and based on this, personalised short video recommendation can be made to provide content that better meets users' tastes and improve user satisfaction and loyalty. Short video recommendation can avoid the user's difficulty in choosing among the huge amount of content and save the user's time and energy in

searching. Intelligent recommendation through artificial neural network technology can better push short videos that meet users' interests [5] and improve user experience. The short video platform through the intelligent recommendation system can accurately understand user needs, accurately match the content of interest for users, improve user activity, increase user viscosity, and then prompt more users to upload and share short video content, forming a virtuous cycle. Personalised short video recommendation helps to increase the user click rate and viewing duration, increase the exposure and click rate of advertisement placement, and thus enhance the commercial value of the platform. In addition, data analysis and mining by artificial neural network technology can help the platform to conduct user profiling and market analysis, and optimize the business model and advertising strategy [6, 7]. Adopting artificial neural network technology for short video recommendation can provide personalised and intelligent recommendation services, improve user experience and platform commercial value, and is one of the important trends in the development of current short video platforms.

Artificial neural network technology can process massive data in real time and quickly and accurately recommend short video content suitable for users. Through parallel computing and optimisation algorithms [8, 9], the efficiency of the short video recommendation system can be improved so that users can instantly access the content they are interested in. Short video recommendation through artificial neural network technology can help platforms to promote and recommend new and high-quality short video content. For up-and-coming video content creators or less noticed videos, the intelligent recommendation system can show these contents to users according to their preferences and interests, increasing their exposure and opportunities.

Artificial neural network technology can analyse and mine a large amount of user behavioural data, such as user viewing duration, preference tags, likes and comments, etc. [10, 11], and through deep learning and pattern recognition of these data, it can discover the hidden needs and behavioural patterns of users and provide powerful support for platform operation decision-making. Artificial neural network technology has a wide range of application prospects and value in short video recommendation, and through continuous optimisation and innovation, it can further improve the accuracy and degree of personalisation of short video recommendation, and provide users with better viewing experience and platform value. Therefore, the research objective of this work is to use artificial neural network technology to quickly obtain the videos that users are interested in from a large number of short videos and save the time of searching video resources.

1.1. **Related Work.** In the last decade, the research on recommender systems has entered into a deeper level, with a large number of algorithms and their derivatives being proposed, there is a demand for higher performance of the algorithms and their applicability in scenarios with large-scale data in industry. The current research status of video recommendation technology is very active and involves several directions, including content-based recommendation, collaborative filtering-based recommendation, deep learning-based recommendation, etc. [12].

Content-based recommendation methods make recommendations for users by analysing the content attributes of the video, such as tags, descriptions, keywords, etc. [13]. This method is able to provide other videos that are similar to its content based on what the user is currently watching or has already liked. Spolaôr et al. [14] reviewed the applications and research on different aspects of content-based video recommendation methods including feature extraction, similarity metrics, recommendation algorithms, etc. and gave a comparison of different feature extraction and similarity metrics methods.

Collaborative filtering-based recommendation methods use users' historical behavioural data and other users' behavioural information, such as viewing records, ratings, preferences, etc., to predict the videos that users may like. This approach can make recommendations based on the similarity of interests between users and can tap into the implicit patterns of user preferences. Chen et al. [15] introduced the principles, methods and applications of collaborative filtering recommender systems, covering the traditional similar user-based and similar item-based collaborative filtering algorithms. Deep learning based recommendation methods utilise deep neural networks to mine the complex relationships between users and videos. This approach can provide more accurate and personalised recommendations by learning models to capture richer features and hidden associations. Da'u and Salim [16] introduced different types of deep learning models and network structures in recommender systems, including deep neural networks, convolutional neural networks, recurrent neural networks and so on.

The data sparsity problem is a major challenge for recommender systems. As the number of users and videos in the online video service platform is huge, it leads to the data matrix is extremely sparse, which is not conducive to producing high quality recommendation results. To alleviate the data sparsity problem, Duma and Twala [17] proposed to add human features in recommending items to alleviate the data sparsity problem in collaborative filtering algorithms. Dvir et al. [18] proposed to cluster the users or videos to reduce the dimensions of the users or videos through clustering, which in turn expands the intersections between the users or videos. Huo et al. [19] proposed using models such as RNN and LSTM to transform the recommendation problem into a sequence prediction problem, which automatically learns the data features, saves a lot of time and performs well.

1.2. **Motivation and contribution.** Due to the limited memory capacity and constant weights of models such as RNN and LSTM, deep learning-based short-frequency recommendation methods do not perform well when dealing with a large number of short video sequences. However, the structure of Attention mechanism (AM) [20, 21] is not limited by the length and number of sequences.

In short video sequences, some frames or segments may be unimportant or contain noisy information.AM can help the model selectively ignore these unimportant contents, thus reducing the interference of noise and improving the robustness and efficiency of the model. Since short video sequences usually contain multiple frames or segments, traditional models may have difficulty in capturing long-term dependencies. In contrast, AM can handle long-term dependencies by adaptively assigning attention weights, enabling the model to better understand the importance and contribution of different moments in the sequence.

Therefore, deep learning algorithms can better achieve complex multi-dimensional feature classification and can adapt to large-scale sample feature analysis. The attention mechanism has obvious advantages in sample feature extraction and analysis, and both of them have high applicability to short video recommendation. Therefore, this paper adopts the Recurrent neural network (RNN) algorithm in deep learning for short video recommendation, and uses AM for feature selection to improve the accuracy of recommendation.

The main innovations and contributions of this work include:

(1) Aiming at the short video recommendation based on the traditional RNN algorithm, there is the problem of constant weights, which leads to large noise interference, a proposed short video recommendation algorithm based on the AM-RNN algorithm is used, which makes the output of the RNN adaptively learnt to the weights.

(2) Aiming at the problem that invalid features in RNN may bring noise, the importance of different second-order features is distinguished by the attention mechanism. The second-order feature vectors and the vectors of the target film are equally sliced, and their relationships are learnt in different hidden semantic spaces at a fine-grained level through the attention mechanism, which improves the weight smoothing problem among the vector elements, and thus improves the performance of the AM-RNN algorithm.

(3) Experimental analyses are conducted for key metrics that affect the performance of the AM-RNN algorithm, such as the regularisation parameter λ, and compared with other recommendation algorithms in terms of RMSE and MAE metrics.

2. **Attention Mechanism.** AM aims at the deep mining of focused features and discards the ineffective training of non-focused features [22]. The demand for feature mining is satisfied by efficient computation of focused features, while filtering out ineffective feature training and reducing the computational complexity. The attention mechanism mainly operates between the query (Q), keywords (K), and weights (V). The mathematical description of the attention mechanism is given below.

Let the total number of system K be L. The degree of similarity between Q and all K_i is first calculated as Similarity(Q, K_i) $(i = 1, 2, \ldots, L)$, and the similarity result is recorded as a score Score(Q, K_i), which is calculated differently because of the differences in the chosen models [23].

(1) Bilinear:

$$\text{Score}(Q, K_i) = K_i W Q \tag{1}$$

(2) Dot product:

$$\text{Score}(Q, K_i) = K_i \cdot Q \tag{2}$$

(3) Scaling dot product:

$$\text{Score}(Q, K_i) = \frac{K_i \cdot Q}{\sqrt{d}} \tag{3}$$

where d denotes the feature dimension and W denotes the linear variable.

Compared with bilinear and dot product, the computational complexity of the scaled dot product model has increased somewhat, but the model has higher resolution, which is beneficial for extracting word features for short video samples. Therefore, the scaled dot product approach is chosen in this study. Let V_i denote the weight value of the i-th K, which is calculated as follows:

$$V_i = \frac{\exp(Score(Q, K_i))}{\sum_{j=1}^{L} \exp(Score(Q, K_j))} \tag{4}$$

Based on the value of V_i , the results of calculating the attention mechanism are shown below [24]

$$Attention(Q, K, V) = \sum_{i=1}^{L} V_i K_i \tag{5}$$

3. **Recurrent Neural Networks (RNN).** The main difference between RNN and ordinary neural networks is reflected in the fact that the output of the network is related to historical inputs and historical hidden layer outputs. The influence of historical data features on the continuity of the current time period is reflected to a larger extent by the influence of historical inputs on the current output. Let x and o be the inputs and hidden layer outputs respectively, the core structure of the RNN recurrent structure is shown in Figure 1. U, V and W are the weights.

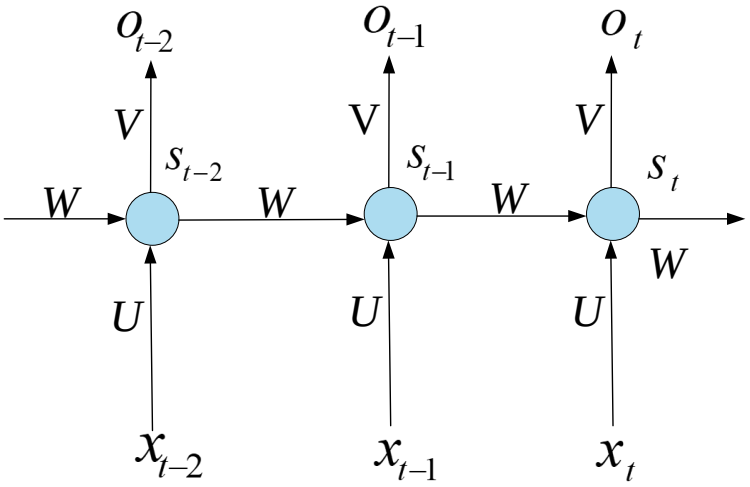

Figure 1. RNN cycle structure

The output at moment t in Figure 1 is related to both moment $t-1$ and moment $t-2$. According to practical needs, the hidden layer loop size can also be increased to expand to more previous historical moments. The training impact of this temporal superposition is more capable of retaining the contextual information of the training samples, thus obtaining more accurate training results, which is precisely the reason why RNN is superior to ordinary neural networks [25].

Let n samples x_i (where $i = 1, 2, \ldots, n$) pass through the hidden layer of RNN to get the input $f(x)$.

$$f(x) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b = \sum_{i=1}^{n} w_ix_i + b \tag{6}$$

where w_i denotes the connection weights of all sample points to the hidden layer and b denotes the bias. Enter $f(x)$ into the conversion function $G(\cdot)$.

$$g(x) = G\left(\sum_{i=1}^{n} w_ix_i + b\right) \tag{7}$$

From Figure 1, the output s_t of the RNN at moment t is related to s_{t-1} and x_t. The s_t is obtained by stimulating $f(\cdot)$ through the hidden layer.

$$s_t = f(Ux_t + Ws_{t-1} + h_t) \tag{8}$$

where h_t is the bias of the excitation at moment t.

The $s(t)$ can be obtained through the output of softmax function.

$$o_t = \text{softmax}(Vs_t + b_0) \tag{9}$$

where b_0 is the output bias.

Regarding U, V, and W solving for RNNs, two approaches can be used, forward and reverse iteration, whereas the former is specific to RNNs and the latter is a common

solving approach for NNs [26].

$$\begin{aligned} o_t &= g(Vs_t) = Vf(Ux_t + Ws_{t-1}) = Vf(Ux_t + Wf(Ux_{t-1} + Ws_{t-2})) \\ &= Vf(Ux_t + Wf(Ux_{t-1} + Wf(Ux_{t-2} + Ws_{t-3}))) \\ &= Vf(Ux_t + Wf(Ux_{t-1} + Wf(Ux_{t-2} + Wf(\mathbf{U}x_{t-3} + \cdots)))) \end{aligned} \tag{10}$$

The constants h_t and b_0 are filtered during the iterative process. By continuously accumulating the calculations, based on the input and output values of the samples, then the U, V and W values can be obtained to determine the stable RNN structure.

Reverse iteration is mainly achieved by continuously reducing the error value δ_k, δ_k can be obtained from the actual value d_k of sample k and the result y_k of RNN training.

$$\delta_k = (d_k - y_k)y_k(1 - y_k) \tag{11}$$

The weights between the hidden layer nodes h_j and y_k are updated as follows:

$$\Delta w_{jk}(n) = \eta(\Delta w_{jk}(n-1) + 1)\delta_k h_j \tag{12}$$

where η is the learning rate.

After updating based on $\Delta w_{jk}(n)$, the latest weight value $w_{jk}(n+1)$ between nodes h_j and y_k is obtained as follows:

$$w_{jk}(n+1) = w_{jk}(n) + \Delta w_{jk}(n) \tag{13}$$

The bias update method for the hidden layer is shown as follows:

$$\Delta b_k(n) = \alpha(\Delta b_k(n-1) + 1)\delta_k \tag{14}$$

where α is the bias update rate.

Get the latest bias $b_k(n+1)$ value based on $\Delta b_k(n)$.

$$b_k(n+1) = b_k(n) + \Delta b_k(n) \tag{15}$$

Calculate the error E.

$$E = \frac{1}{2}\sum_{k=1}^{M}(d_k - y_k)^2 \tag{16}$$

where M denotes the total number of output nodes.

Since Equation (10) filters out the biased iterations in the computation, the main part of the computation is the solution of U, V and W , which is equivalent to the forward iteration that only performs the updating of some of the parameters of the RNN. However, the reverse iteration needs to update all the parameters of the RNN, so the latter outperforms the former in terms of the completeness of the solved model parameters. However, the efficiency and complexity of inverse solving is significantly higher than the former. In practical use, the RNN parameter solving method is selected according to demand. Since the main purpose of short video recommendation in this study is to improve the recommendation accuracy, forward iteration is used.

4. **Short video recommendation algorithm based on AM-RNN algorithm.**

4.1. **Basic framework of AM-RNN.**

The basic framework of the AM-RNN algorithm designed in this work is shown in Figure 2, including (1) sparse feature data input layer: (2) embedding layer; (3) RNN model layer; (4) AM layer; and (5) output layer.

(1) Sparse feature data input layer. Usually the original data in the recommendation domain has the problem of sparse data, such as user ID, item ID, etc., but at this time these vectors are extremely sparse and the length of the vectors is generally very long. These factors lead to the fact that these encoded features cannot be directly input into the model for training, and can lead to problems such as too many model parameters and

difficulty in convergence. Therefore, it is necessary to encode these features one-hot first [27], so that each user and each item is characterised by a unique vector.

(2) Embedding layer. To address the problem in (1), before data input, the encoded feature vectors need to be transformed into continuous vectors of appropriate length through the mapping matrix to alleviate the problems caused by the sparsity of the data. The embedding layer is actually an initialised matrix, and the mapping process is essentially a matrix multiplication as shown in Figure 3. The left side of the equal sign consists of two parts: the sparse vector after one-hot coding and the initialised matrix of the embedding layer. The sparse vector and the embedding layer matrix are multiplied into a matrix subscript selection, and the initialised matrix will be updated during the training process, so the result of the multiplication is constantly updated.

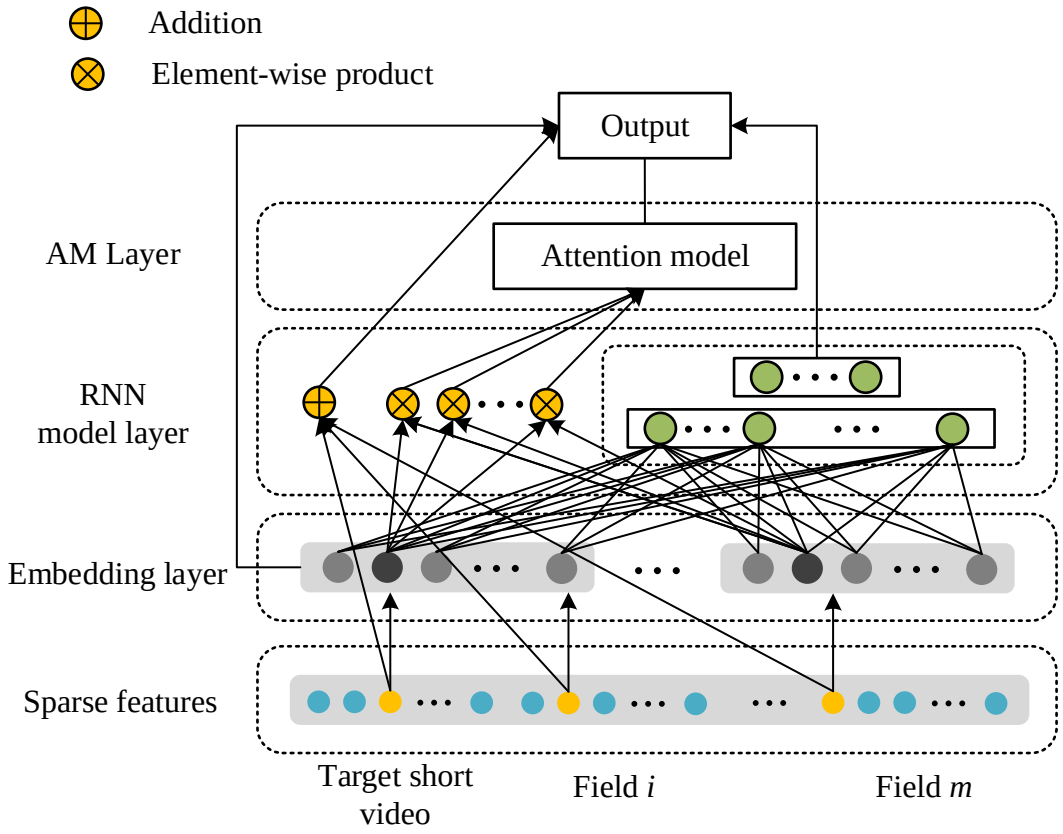

Figure 2. Basic framework of the AM-RNN algorithm

$$[0 \quad 1 \quad 0 \quad 0 \quad 0] \times \begin{bmatrix} 7 & 5 & 2 & 9 \\ 1 & 7 & 4 & 8 \\ 3 & 9 & 0 & 3 \\ 1 & 5 & 7 & 6 \\ 2 & 9 & 3 & 6 \end{bmatrix} = [1 \quad 7 \quad 4 \quad 8]$$

Figure 3. Embedding layer matrix multiplication

(3) RNN model layer. In RNN model, the set of all features expressed in the embedding layer is denoted by $u^{(0)}$. The output of the DNN model is shown as follows:

$$u^{(0)} = [e_1, e_2, e_3, \ldots, e_n] \tag{17}$$

$$y_{RNN} = \sigma(W^{(H+1)} \cdot u^{(H)} + b^{(H+1)}) \tag{18}$$

where e_i denotes the embedding layer expression of the i-th word feature, and there are n word features in total. $w^{(H+1)}$ and $b^{(H+1)}$ denote the weights and biases of the $(H+1)$-th layer of the network's implicit layer, σ denotes the activation function, and y_{RNN} denotes the output of the RNN model.

(4) AM Layer: The AM model is able to make a careful selection of word features, reducing the weight of features that do not contribute to the recommendation task and increasing the weight of word features that do contribute to the recommendation task. Note that the output of the AM model through the attention module is y_{AM} . How the module generates y_{AM} will be discussed in detail in the next subsection.

(5) Output layer. The output module fuses the adaptive weight output of the AM model and the output of the RNN model as follows:

$$y = \sigma_o\left(concat(y_{AM}, y_{RNN})\right) \tag{19}$$

where σ_o denotes the activation function of the output layer.

4.2. **Objective function.** The AM layer has two inputs, the word feature vector and the vector of the target video. The purpose of using the attention mechanism in this step is to further dig into which part of the vector of features is more helpful for the prediction of the target video rating. The elemental product of the features is just computed before inputting into the AM model, which is done on the one hand to keep the same dimension size of each word feature and the embedding layer representation of the target video to facilitate the subsequent computation of the attention weights, and on the other hand to portray the output weights of the attention mechanism at a finer granularity.

Typically, the output of an AM-based model can be described as follow [28]:

$$A = \sum_{i=0}^{n} a_i \mathbf{v}_i \tag{20}$$

where a_i denotes the input vector $\mathbf{v}_i$, learnt weights after the AM model, and the vector length is n.

It can be seen that each element in the vector $\mathbf{v}_i$ is multiplied by an identical value a_i, and in this way the attentional weights are computed. This makes it difficult to ensure that a portion of the elements in the vector $\mathbf{v}_i$, that have a higher contribution to the prediction result, are not influenced by the other remaining elements that have a lower contribution. In order to reduce this influence, the vectors of the two inputs of the AM are sliced into h parts of equal length, thus learning the attention weights of each of these h equal-length vectors at a finer granularity. At this point the output of the Attention Mechanism model can be described as:

$$A = \sum_{i=0}^{n} \sum_{m=0}^{h} a_{im} \mathbf{v}_{im} \tag{21}$$

where a_{im} denotes the attentional weight of the mth equal-length vector $\mathbf{v}_{im}$.

Firstly, the data belonging to the category type needs to be one-hot coded and the data belonging to the continuous type needs to be normalized before data input. Secondly, this processed sparse data is passed through the embedding layer. This step is essentially a matrix multiplication operation. The embedding layer vectors are also divided equally into h parts and mapped to the nonlinear hidden semantic space via a nonlinear activation function, which is similar to the nonlinear activation function in deep learning networks

used to mine the nonlinear relationships between data. The vector mapping of the target short video is shown as follow:

$$S_{nk} = \mathcal{F}_k(w_{nk}) \quad n = 1, 2, \cdots, N \quad k = 1, 2, \cdots, h \tag{22}$$

$$S_{tk} = \mathcal{F}_k(w_k), \quad k = 1, 2, \ldots, h \tag{23}$$

where w_{nk} denotes the k-th hidden semantic vector after the vector w_n has been homogenised, similarly, w_k denotes the k-th hidden semantic vector after the vector w_t has been homogenised, and $\mathcal{F}_k$ denotes the nonlinear mapping function of the k-th vector.

Then, the scaled inner product is used to calculate the attention weights, as shown in Figure 4. It is assumed that A and B denote the hermitian semantic space matrix of all word feature vectors and the hermitian semantic space of the target short video vectors, respectively. The A and B vectors need to be further scaled after the inner product operation in order to prevent vectors with too long inputs that may lead to an excessively large size of the required inner product, which may result in the subsequent solution of the softmax when obtaining very small gradients.

$$S_k = softmax\left(\frac{S_{nk}{S_{tk}}^{\mathrm{T}}}{\sqrt{d_r}}\right) S_{nk}, \quad n = 1, 2, \cdots, N \quad k = 1, 2, \cdots, h \tag{24}$$

where d_r denotes the length of the input vector.

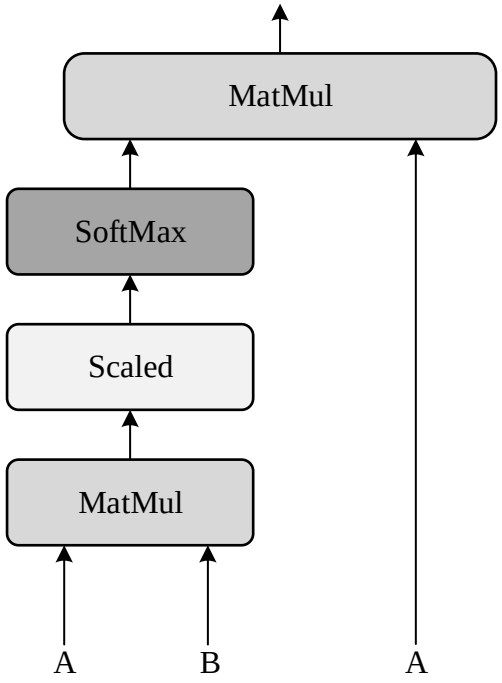

Figure 4. Scaled Dot-product Attention

After two matrix multiplications, the dimension of the output S_k is $h \times d_r$, and the process is equivalent to performing an accumulation operation on all word features of the k-th copy of the hidden semantic space.

The output of Equation (24) is the result of computing the attention of the original word feature vectors in the h cryptosemantic space, which therefore needs to be merged into the full semantic space as follows:

$$S = \mathrm{concat}(s_1, s_2, \ldots, s_h) \tag{25}$$

So the output of the AM layer is shown as follow:

$$y_{AM} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{m=0}^{h \times d_r} w_m \mathbf{S} \tag{26}$$

where w_0 is the global offset, w_i is the weight of the word feature, and w_m is the weight of each element in the vector $\mathbf{S}$.

The final output of the AM-RNN algorithm is shown as follow:

$$y_{AM-RNN} = y_{AM} + y_{RNN} \tag{27}$$

Since the AM-RNN algorithm employs two evaluation metrics, RMSE and MAE, its objective function also exists in two forms. The objective function using the RMSE evaluation metric is shown in Equation (28). The objective function using MAE evaluation metrics is shown in Equation (29). In order to prevent the model from overfitting, regularization terms were introduced to both objective functions and the objective function was optimised using the Adam optimiser.

$$L_{RMSE} = min \sqrt{\frac{1}{|N|} \sum_{(u,i) \in N} \left(y_{AM-RNN} - y_{ui}\right)^2} + \lambda \|\mathbf{W}\|^2 \tag{28}$$

$$L_{MAE} = min \frac{1}{|N|} \sum_{(u,i) \in N} |y_{AM-RNN} - y_{ui}| + \lambda \|\mathbf{W}\|^2 \tag{29}$$

where u denotes the user, i denotes the movie, N denotes the test set, y_{AM-RNN} denotes the AM-RNN algorithm predicts the rating of user u for movie i, y_{ui} denotes the true rating of user u for movie i, $\lambda \|\mathbf{W}\|^2$ is the regularization term, λ is the regularization parameter, and $\mathbf{W}$ denotes the output layer weight matrix.

5. Experimental results and analyses.

5.1. Effect of different word vector sizes.

In order to verify the performance of AM-RNN algorithm in short video recommendation, user playback history data of Tencent video is used. Tencent video has a large user base, with more than 230 million monthly active users, and more than 30 million users online at the same time during the daily peak hours. Log data from 100,000 users and 325,128 short videos totalling about 700,000 playback records in 15 days were randomly selected. The simulation dataset was classified into four types, films, animation, news and sports, as shown in Table 1. The simulation experiments were conducted in a multi-core PC host with 32G RAM, 8-core 3.4GHz CPU, and NVIDIA GeForce GTX 3090 GPU.

Table 1. Simulation data set.

Typology	Sample size	Formatting
Film	1308	wmv, avi
Anime	1316	wmv, avi
News	1401	MP4
Sport	1023	MP4

In the 4-class sample set, the length of the description text of the short video varies greatly, and the number of word feature volume it produces varies significantly through word splitting, respectively, different scales of word features are selected for AM-RNN prediction performance simulation, in which AM adopts the full-window mode, and all

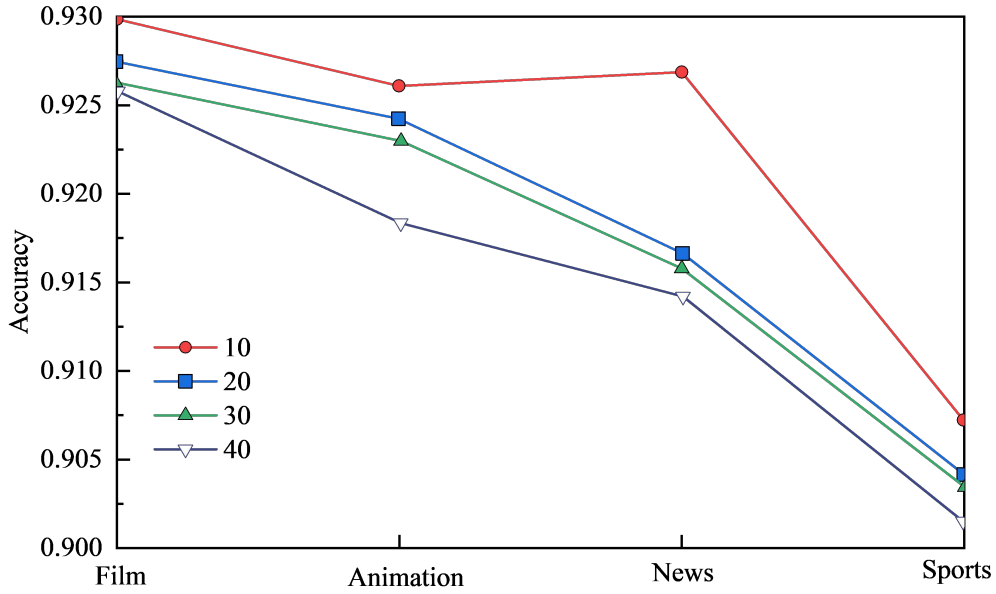

Figure 5. Recommendation Accuracy for Different Word Vector Sizes

the word features are included in the weighted summation. The recommendation accuracy at different word vector sizes is shown in Figure 5.

As shown in Table 2, for the same sample set of short videos, the classification accuracy of AM-RNN algorithm under different word feature sizes has a small difference. Comparing the classification accuracy of four types of short videos under different word feature sizes, it is found that there is a slight decrease in the accuracy rate after the word feature size is increased, which indicates that AM-RNN has a higher stability of sentiment classification on microblogging data of different sizes. Comparing horizontally, the AM-RNN algorithm has the best recommendation accuracy on films with a range of about [0.9021,0.9054], while the recommendation accuracy on sports is lower.

It can be seen that under the condition of the same sample set, the recall and F1 value are not sensitive to the change of word feature size, which indicates that the AM-RNN algorithm has a high applicability to the 4-class dataset. In the 4-class dataset, the film dataset has the best performance, while the sports dataset has the worst performance, which may be related to the video format and the number of samples.

5.2. **Performance with different attention window sizes.** The attention window size determines the number of word features in the process of forming features by weighted summation of word features using AM. Based on the proportion of word features to the total word vectors α, the recommendation readiness and recommendation time of the AM-RNN algorithm are determined as shown in Table 3.

Choosing different attention window sizes has a large impact on the short video recommendation accuracy and efficiency of the AM-RNN algorithm. For the same sample set, the AM-RNN algorithm has the highest recommendation accuracy at $\alpha = 100\%$, i.e., when all the word features are involved in weighting to sentence vectors. The smaller the value of α, the lower the recommendation accuracy, which also illustrates the close connection between accuracy and the completeness of the predicted partition. In comparison, it is found that when $\alpha = 80\%$, the recommendation accuracy of AM-RNN is not the highest, but it is very close to the accuracy corresponding to $\alpha = 100\%$. In terms of recommendation time, the larger the value of α is, the more word feature vectors are

Table 2. Recall and F1 values for different word feature sizes.

Data set	Number of word features	Recall rate	F1 value
Film	10	0.9054	0.9015
	20	0.9046	0.902
	30	0.9043	0.9052
	40	0.9021	0.8946
Anime	10	0.9011	0.8911
	20	0.8997	0.8943
	30	0.8974	0.0828
	40	0.8966	0.8904
News	10	0.8958	0.8895
	20	0.8952	0.8926
	30	0.893	0.8913
	40	0.8911	0.8866
Sport	10	0.8912	0.8818
	20	0.8901	0.888
	30	0.8867	0.8862
	40	0.872	0.8632

Table 3. Recommended Performance for Different Attention Windows.

Data set	$\alpha\%$	Accuracy	Classification time/s
Film	20	0.8271	1.3269
	40	0.886	2.2249
	60	0.8942	3.4719
	80	0.9117	4.2289
	100	0.9196	6.5279
Anime	20	0.8163	1.3049
	40	0.8796	2.4589
	60	0.89	3.6119
	80	0.9097	4.8209
	100	0.9157	6.7859
News	20	0.8175	1.3929
	40	0.8832	2.5499
	60	0.8923	3.5779
	80	0.9104	4.7049
	100	0.9166	6.5829
Sport	20	0.799	1.6519
	40	0.8223	2.7139
	60	0.8551	3.6689
	80	0.8894	4.9019
	100	0.897	6.8059

involved in the computation in AM, and the longer the recommendation time is. Therefore, in order to balance the recommendation accuracy and time, the size of the attention

window should be set reasonably, and $\alpha = 80\%$ is chosen in the subsequent simulation experiments.

5.3. **Ablation experiment analysis.** In order to verify the optimisation performance of AM for RNN, ablation simulation experiments are conducted and the results are shown in Figure 6.

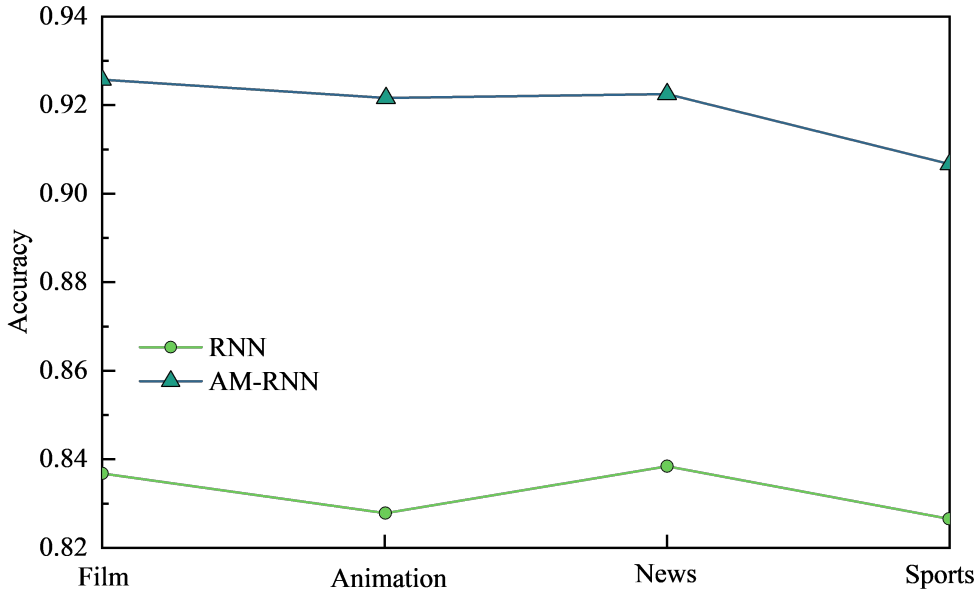

Figure 6. Classification accuracy of RNN and AM-RNN algorithms

It can be seen that the AM-RNN algorithm improves the recommendation accuracy significantly for the four types of short videos compared with RNN.AM-RNN are all above 0.9, while the highest value of RNN is only 0.84, so after the introduction of the AM model, it has a significant effect on the improvement of the performance of the RNN.The recall and F1 of the RNN and the AM-RNN are shown in Table 4.

Table 4. Recall of RNN vs. AM-RNN and F1

Data set	Arithmetic	Recall rate	F1 value
Film	RNN	0.8078	0.7941
	AM-RNN	0.9071	0.8922
Anime	RNN	0.8033	0.7852
	AM-RNN	0.9024	0.8907
News	RNN	0.8054	0.7908
	AM-RNN	0.9042	0.891
Sport	RNN	0.8065	0.7754
	AM-RNN	0.8974	0.8711

5.4. **Effect of regularisation parameters.** Since the training process of AM models may lead to overfitting, and also the training of RNN models may lead to overfitting, adding a regularization term to the output layer of AM-RNN can help to prevent overfitting, and also make the training results more robust.

The setting of the regularization parameter λ will directly affect the overall performance of the AM-RNN. Table 5 lists the results of the AM-RNN algorithm achieving the optimal RMSE and MAE under different λ is, where the hidden semantic space parameter $h = 8$.

Table 5. RMSE and MAE for different values of λ

λ	**RMSE**	**MAE**
0.02	0.8632	0.67633
0.004	0.8592	0.67263
0.002	0.8541	0.66643
0.0004	0.8533	0.66472
0.0002	0.8535	0.66513

It can be seen that as the value of λ decreases, the results of the two evaluation indicators, RMSE and MAE, show a decreasing and then increasing trend, and both RMSE and MAE at $\lambda = 0.0004$ reach the lowest value, so the model is most effective when $\lambda = 0.0005$ is taken.

5.5. **Performance comparison of different algorithms.** In order to compare the performance of the AM-RNN algorithm and commonly used recommendation algorithms, SVM [29], Capsule network (CN) [30], PSO-LSTM [31], and AM-RNN are used for example simulation respectively, and the simulation results are shown in Table 6.

Table 6. Short Video Recommendation Performance of 4 Algorithms

Typology	**Arithmetic**	**Accuracy**	**Recall rate**	**F1**
Film	SVM	0.837	0.8122	0.8064
	CN	0.8918	0.8872	0.8804
	PSO-LSTM	0.9004	0.8952	0.8825
	AM-RNN	0.9196	0.9071	0.8922
Anime	SVM	0.8092	0.7865	0.7746
	CN	0.8857	0.8772	0.8657
	PSO-LSTM	0.8872	0.8791	0.8642
	AM-RNN	0.9157	0.9024	0.8907
News	SVM	0.8148	0.7945	0.7826
	CN	0.8919	0.8757	0.8694
	PSO-LSTM	0.8992	0.8832	0.8763
	AM-RNN	0.9166	0.9042	0.891
Sport	SVM	0.7824	0.7712	0.7554
	CN	0.8772	0.8624	0.8338
	PSO-LSTM	0.8801	0.8698	0.8365
	AM-RNN	0.897	0.8974	0.8711

6. **Conclusion.** In this work, RNN algorithm in deep learning is used for short video recommendation, and with the help of with AM for feature selection to improve the accuracy of recommendation. Firstly, for the short video recommendation based on traditional RNN algorithm has the problem of constant weights, which leads to large noise interference, a proposed short video recommendation algorithm based on AM-RNN algorithm is used, which makes the output of RNN adaptively learn the weights. Secondly, the importance of different second-order features is distinguished through the attention mechanism. The performance of the AM-RNN algorithm is improved by equally slicing the second-order feature vectors and the vectors of the target film, learning their relationships in different hidden semantic spaces at a fine-grained level through the attention mechanism,

and improving the weight smoothing problem among the vector elements. Experimental results show that the AM-RNN algorithm significantly outperforms other recommendation algorithms in terms of RMSE and MAE metrics. However, there is a cold-start problem for both new users and new videos, and future work will try to apply the AM-RNN algorithm to data with user intrinsic attributes and video intrinsic attributes to further solve the cold-start problem.

REFERENCES

[1] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of Quantum Genetic Optimization of LVQ Neural Network in Smart City Traffic Network Prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[2] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, 40, 2019.

[3] F. Zhang, T.-Y. Wu, and G. Zheng, "Video salient region detection model based on wavelet transform and feature comparison," *EURASIP Journal on Image and Video Processing*, vol. 2019, 58, 2019.

[4] C. Troussas, A. Krouska, and C. Sgouropoulou, "Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education," *Computers & Education*, vol. 144, 103698, 2020.

[5] Y. Li, C. Chen, X. Zheng, Y. Zhang, B. Gong, J. Wang, and L. Chen, "Selective and collaborative influence function for efficient recommendation unlearning," *Expert Systems with Applications*, vol. 234, 121025, 2023.

[6] M. C. V. Joe, and D. J. S. Raj, "Location-based orientation context dependent recommender system for users," *Journal of Trends in Computer Science and Smart Technology*, vol. 3, no. 1, pp. 14-23, 2021.

[7] P. Xu, K. Wang, M. M. Hassan, C.-M. Chen, W. Lin, M. R. Hassan, and G. Fortino, "Adversarial Robustness in Graph-Based Neural Architecture Search for Edge AI Transportation Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8465-8474, 2023.

[8] K. Wang, Z. Chen, X. Dang, X. Fan, X. Han, C.-M. Chen, W. Ding, S.-M. Yiu, and J. Weng, "Uncovering Hidden Vulnerabilities in Convolutional Neural Networks through Graph-based Adversarial Robustness Evaluation," *Pattern Recognition*, vol. 143, 109745, 2023.

[9] H. Peng, S. Ma, and J. M. Spector, "Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment," *Smart Learning Environments*, vol. 6, no. 1, pp. 1-14, 2019.

[10] O. Tyrväinen, H. Karjaluoto, and H. Saarijärvi, "Personalization and hedonic motivation in creating customer experiences and loyalty in omnichannel retail," *Journal of Retailing and Consumer Services*, vol. 57, 102233, 2020.

[11] K. Sailunaz, and R. Alhajj, "Emotion and sentiment analysis from Twitter text," *Journal of Computational Science*, vol. 36, 101003, 2019.

[12] A. Rai, "Explainable AI: From black box to glass box," *Journal of the Academy of Marketing Science*, vol. 48, pp. 137-141, 2020.

[13] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? A survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, pp. 813-831, 2019.

[14] N. Spolaôr, H. D. Lee, W. S. R. Takaki, L. A. Ensina, C. S. R. Coy, and F. C. Wu, "A systematic review on content-based video retrieval," *Engineering Applications of Artificial Intelligence*, vol. 90, 103557, 2020.

[15] Y.-C. Chen, L. Hui, and T. Thaipisutikul, "A collaborative filtering recommendation system with dynamic time decay," *The Journal of Supercomputing*, vol. 77, pp. 244-262, 2021.

[16] A. Da'u, and N. Salim, "Recommendation system based on deep learning methods: a systematic review and new directions," *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2709-2748, 2020.

[17] M. Duma, and B. Twala, "Sparseness reduction in collaborative filtering using a nearest neighbour artificial immune system with genetic algorithms," *Expert Systems with Applications*, vol. 132, pp. 110-125, 2019.

[18] A. Dvir, A. K. Marnerides, R. Dubin, N. Golan, and C. Hajaj, "Encrypted video traffic clustering demystified," *Computers & Security*, vol. 96, 101917, 2020.

[19] Y. Huo, D. F. Wong, L. M. Ni, L. S. Chao, and J. Zhang, "Knowledge modeling via contextualized representations for LSTM-based personalized exercise recommendation," *Information Sciences*, vol. 523, pp. 266-278, 2020.
[20] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu, "Attention mechanisms in computer vision: A survey," *Computational Visual Media*, vol. 8, no. 3, pp. 331-368, 2022.
[21] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48-62, 2021.
[22] W. Li, K. Liu, L. Zhang, and F. Cheng, "Object detection based on an adaptive attention mechanism," *Scientific Reports*, vol. 10, no. 1, 11307, 2020.
[23] J. Li, K. Jin, D. Zhou, N. Kubota, and Z. Ju, "Attention mechanism-based CNN for facial expression recognition," *Neurocomputing*, vol. 411, pp. 340-350, 2020.
[24] C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, and Q. Dai, "STAT: Spatial-temporal attention mechanism for video captioning," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 229-241, 2019.
[25] X. Zhang, C. He, Y. Lu, B. Chen, L. Zhu, and L. Zhang, "Fault diagnosis for small samples based on attention mechanism," *Measurement*, vol. 187, 110242, 2022.
[26] W. Cai, B. Zhai, Y. Liu, R. Liu, and X. Ning, "Quadratic polynomial guided fuzzy C-means and dual attention mechanism for medical image segmentation," *Displays*, vol. 70, 102106, 2021.
[27] Y. Chen, G. Peng, Z. Zhu, and S. Li, "A novel deep learning method based on attention mechanism for bearing remaining useful life prediction," *Applied Soft Computing*, vol. 86, 105919, 2020.
[28] X. Li, W. Zhang, and Q. Ding, "Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism," *Signal Processing*, vol. 161, pp. 136-154, 2019.
[29] M. Mohammadi, T. A. Rashid, S. H. T. Karim, A. H. M. Aldalwie, Q. T. Tho, M. Bidaki, A. M. Rahmani, and M. Hosseinzadeh, "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems," *Journal of Network and Computer Applications*, vol. 178, 102983, 2021.
[30] V. Mazzia, F. Salvetti, and M. Chiaberge, "Efficient-capsnet: Capsule network with self-attention routing," *Scientific Reports*, vol. 11, no. 1, 14634, 2021.
[31] V. Gundu, and S. P. Simon, "PSO–LSTM for short term forecast of heterogeneous time series electricity price signals," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 2375-2385, 2021.