

LAR: A User Behavior Prediction Model in Server Log Based on LSTM-Attention Network and RSC Algorithm

Yingying Shang^{a1}

^a *School of Computer Engineering and Science, Shanghai University, Shanghai 200444, P.R. China*

Abstract. Using server log data to predict the URLs that a user is likely to visit is an important research area in user behavior prediction. In this paper, a predictive model (called LAR) based on the long short-term memory (LSTM) attention network and reciprocal-nearest-neighbors supported clustering algorithm (RSC) for predicting the URL is proposed. First, the LSTM-attention network is used to predict the URL categories a user might visit, and the RSC algorithm is then used to cluster users. Subsequently, the URLs belonging to the same category are determined from the user clusters to predict the URLs that the user might visit. The proposed LAR model considers the time sequence of the user access URL, and the relationship between a single user and group users, which effectively improves the prediction accuracy. The experimental results demonstrate that the LAR model is feasible and effective for user behavior prediction. The accuracy of the mean absolute error and root mean square error of the LAR model are better than those of the other models compared in this study.

Keywords. Behavior prediction, LSTM, attention mechanism, cluster, server log

1. Introduction

With the development of the Internet, a massive amount of server log information has become a treasure trove for data mining. The server log not only contains information on the running status of the server, but also contains information related to user behavior. The analysis of valuable information from massive log files and prediction of user behavior is very helpful in balancing the network load, protecting young people from harmful information, and identifying potential customers.

A research direction of web data mining involves the development of predictive models of the access behavior a web user, which can be used to predict the next set of pages that the user might access. The objective of this paper is to predict the user's next access behavior through the historical access records generated by web users in the log. Various models such as fuzzy interference models, support vector machines, artificial neural networks (ANNs), association rule mining, and Markov models have been proposed to address the problem of web page prediction. Among them, the Markov

¹ Corresponding Author: Yingying Shang, Shanghai University, China; Email: yogoshary@163.com

model, which is the most commonly used model, uses the access records to calculate the probability of web page conversion, and predicts the next web page the user is likely to visit. B. Harindra Varma et al. [1] combined the naive Bayesian algorithm with the Markov model to predict the next category of user access. Xing Dongshan et al. [2] proposed a mixed-order Markov tree model. Awad et al. [3] studied several hybrid models combining different classification techniques, such as the Markov model, ANN, and Kth-order Markov model, and solved and predicted problems using Dempster rules. Trincherio et al. [4] analyzed and studied the application of the Markov and All-Kth Markov models for web prediction. Panchal et al. [5] studied the Markov model to predict the browsing behavior of users at the category level, and presented a mining process of web server log files to extract usage patterns for web link prediction with the proposed Markov Model. Pandey et al. [6] integrated a low-order Markov model and clustering to predict user behavior.

However, a predictive model based on the Markov model ignores the influence of long-distance behavior on current behavior. Concurrently, a majority of the user behavior prediction models only consider the access behavior of a single user, while ignoring the relationship between a single user and group users. Therefore, by combining the advantages of the long short-term memory (LSTM) network and attention mechanism, in this paper, a predictive model of user behavior from web logs based on LSTM-attention and reciprocal-nearest-neighbors supported clustering (RSC) (named prediction of web-user behavior based on LSTM-attention and RSC, LAR in short) is proposed. The proposed model incorporates the LSTM network [7], attention mechanism [8], and RSC algorithm [9]. The LSTM network structure can solve the problem ignored by the Markov model, i.e., the influence of short-term and long-term behaviors on current behavior. The attention mechanism can compensate for the limitation of the fixed vector length in the codec of the LSTM structure, and help capture important information in the long input [10]. Basic information on a user's access behavior is used as the feature input, which is then trained by the LSTM-attention network. Concurrently, the user is clustered by the RSC algorithm, and the URL prediction of a single user's expected access behavior is made based on the group users' access record. The experimental data presented in this paper are from the private server of a university campus network, and the proposed model is evaluated and analyzed using real server log datasets. The experimental results show that the LAR model can accurately predict the URL that users might visit in the future. In practice, using the LAR model to predict the online behavior of campus network users, the most concerned information that users pay attention to can be determined, and this information can subsequently be used to improve the campus service.

2. Construction of the LAR Model

2.1. Model Framework

The LAR model is shown in Figure 1.

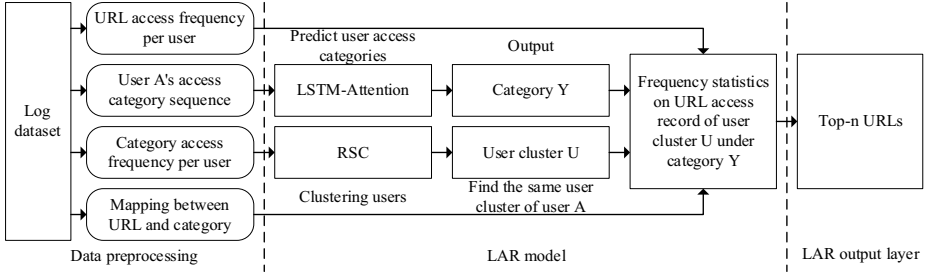


Figure 1. Structure of the LAR model

The proposed model can be divided into three parts, namely the LSTM-attention network, RSC, and section that generates the prediction results. First, the pre-processed data are used as the input to the predictive model. Then, the LSTM-attention model is constructed based on the LSTM network and attention mechanism, and subsequently used to predict the URL category. Concurrently, the RSC algorithm is used to obtain the user clusters. Finally, we use the URL categories predicted by LSTM-attention and the user clusters, combine the mapping relationship between the URL category and URL, and the URL access frequency of each user, to obtain the top N URLs, which is the final result of the LAR model.

The specific steps of the LAR model are: (1) Following normalization, the log data are used as the input to the predictive model. (2) The LSTM-attention network is used to predict the URL category sequence of user A, and the predicted result is category Y. (3) The RSC algorithm is used to cluster users and find the cluster U of user A. (4) Based on the historical access records of user cluster U and mapping relationship between the URL category and URL, the access records with category Y are extracted. Using the frequency statistics of the access records, the top-n URLs with the highest frequency are determined. The top-n URLs constitute the output of the LAR model.

2.2. Model Details

2.2.1 LSTM.

The LSTM network proposed by Hochreiter et al. [7] is an improved model of the recurrent neural network. The LSTM network learns to forget the information from the cell state and updates the information of the cell state via the memory unit. This enables the network to effectively use the long-distance dependent information in the sequence data and solve the problem of vanishing gradient or exploding gradient; thus, this network is mainly used to address time series problems. The structure of each LSTM unit includes a memory cell and three gates. The memory cell is set to record the state of the unit. The input gate and output gate are used to receive the output parameters and modify these parameters, respectively. The forget gate is used to control the forgotten degree of the previous unit state [11]. The structure of the LSTM unit is shown in Figure 2.

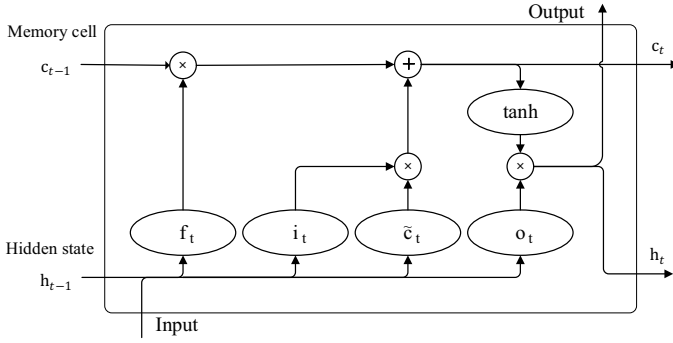


Figure 2. Structure of the LSTM unit

Assuming that the time step is t ($t=1, \dots, n$, where n is the total number of time steps), x_t is the input state of the LSTM, f_t is the forget gate, i_t is the input gate, and o_t is the output gate. Furthermore, c_t is the unit state memory, and h_t is the hidden state output. At the previous time step $t-1$, c_{t-1} represents the memory cell, and h_{t-1} is the hidden state output. The relationships between the variables can be described as:

$$f_t = \text{sigmoid}(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \tag{1}$$

$$i_t = \text{sigmoid}(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \tag{2}$$

$$\tilde{c}_t = \text{tanh}(W_{ch}h_{t-1} + W_{cx}x_t + b_c) \tag{3}$$

$$o_t = \text{sigmoid}(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \tag{4}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \tag{5}$$

$$h_t = o_t \otimes \text{tanh}(c_t) \tag{6}$$

Here, W and b are the training network parameters.

2.2.2. LSTM-Attention Network.

In deep neural networks, two problems might affect the training process. One is that the disappearance of the gradient makes it difficult to achieve neural network convergence, while the other pertains to over-fitting that causes the test set to fail. The LSTM network can effectively solve the gradient disappearance problem, and the dropout layer can alleviate the problem of overfitting by preventing the neurons from co-adapting. Therefore, in the design of the LSTM neural network, a dropout layer is added to optimize the structure of the neural network.

To capture the effective information in the log sequence better and grasp important information, in this paper, an LSTM network structure with the attention mechanism is proposed. This structure overcomes the problem that the standard LSTM model cannot fully learn the details of sequence coding due to the use of the same state vector in each prediction step. The LSTM-attention structure is shown in Figure 3.

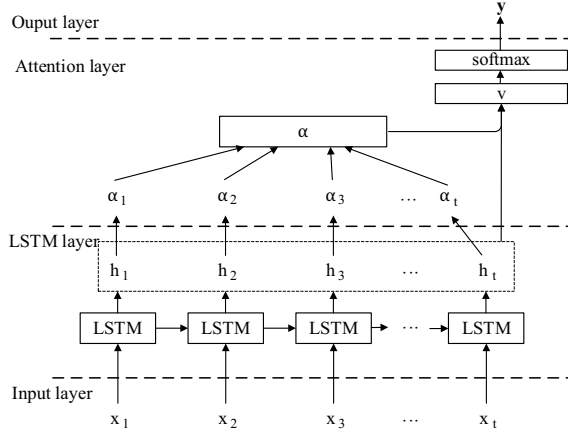


Figure 3. Structure of the LSTM-attention network

In Figure 3, the input sequence (x_1, x_2, \dots, x_t) represents the URL category data obtained in time t . Each item in the input sequence is passed to the LSTM unit to obtain the output (h_1, h_2, \dots, h_t) of the hidden layer. In the hidden layer, the attention mechanism is introduced to calculate the probability distribution values of each input assignment, $(\alpha_1, \alpha_2, \dots, \alpha_t)$. The attention mechanism generates the attention weight matrix α and feature representation v . The corresponding equations are given by:

$$u_i = \tanh(W_s h_i + b_s) \quad (7)$$

$$\alpha_i = \exp(u_i^T u_s) / (\sum_{i=1}^t \exp(u_i^T u_s)), \quad \sum_{i=1}^t \alpha_i = 1 \quad (8)$$

$$v = \sum_{i=1}^t \alpha_i h_i \quad (9)$$

Here, W_s is the weight matrix, b_s is the offset, and u_s is the random initial attention matrix. Finally, the softmax function is used to obtain the predictive category y , which can be calculated as:

$$y = \text{softmax}(W_v v + b_v) \quad (10)$$

2.2.3 RSC Clustering Algorithm.

In this study, the RSC algorithm is used to cluster users, find the top- n URLs of each category of similar users, and improve the accuracy of the single user prediction model. The basic idea of the RSC algorithm is that two nearest data points should be placed in a cluster [9]. In the RSC algorithm, data points are connected with their nearest neighbors to build a clustering tree. Each clustering tree has a pair of points that are each other's closest neighbors, which are known as reciprocal nearest neighbors. The midpoints of the reciprocal nearest neighbors represent the center of the cluster, and can be used for a hierarchical aggregation of unclassified data. According to the previous research, tests on standard datasets across multiple domains show that RSC method is much faster and more accurate than the state-of-the-art benchmarks [9].

The RSC algorithm is used to cluster users according to the access frequency of the users in each URL category. The process is as follows: a) Represent user points with vectors. If the access frequency of user A in category c is N_A^c , then each user's access record can be represented by a 1*f-dimensional vector $V.V_A = [N_A^0, N_A^1, \dots, N_A^{f-1}]^T$, for $\forall i, 0 \leq i \leq n$, where n is the number of user points, and f is the number of categories. b) Calculate the nearest neighbor points of all users, and direct each user point to its nearest neighbor point. Calculate the Euclidean distance between all users. The point with the smallest distance is the nearest neighbor. If y is the nearest neighbor of x, it is recorded as $x \rightarrow y$. c) Find the members and central points of each user cluster. If $x \rightarrow y$ and $y \rightarrow x$, x and y are the nearest neighbor pairs. The midpoint of the nearest neighbor point pair is the center point of a cluster. The point attached to the nearest neighbor point pair is the member of the cluster. The number of points passed by each member to the nearest neighbor point is the depth of the member. In addition, to avoid the appearance of clusters that are too thin and long, when the depth of the member is greater than the threshold T, the member point is disconnected, and it points to the nearest neighbor point that is not in the current cluster. Let the number of points in this cluster be n, and the threshold T is defined as $T = \lceil \log N + 1 \rceil$.

3. Experiments Results and Analysis

3.1. Setup of the Experiment

3.1.1 Data Collection and Processing.

This paper used the desensitized data on a non-public production server of a university to verify the effectiveness of the proposed model. The size of the original log file is 51 GB, with 80 million lines of data. As an important part of user behavior prediction, the log data records all the information in the network requests, including the pages visited by users, server status information, and other kinds of data. While most lines of the log data have a relatively neat format, there are also many lines with no obvious rules, which need to be collected and cleaned using a web crawler. This paper collected the log data of the fourth week of December 2019 (5 days from Monday to Friday). In total, 398,241 log data items were stored in the database after data cleaning, with 198 users and 26 categories. The standardized data contained 8 columns, as shown in Figure 4.

mysql_id	category	host_ip	user_hash	dst_host	record_time	dst_ip	url
19	IT	10.89.4...	304fcc7f...	cpro.bai...	157698533...	112.122...	cpro.bai...
30	Market...	10.88.2...	4b19612...	www.ge...	157751190...	39.107...	www.ge...
42	Video ...	10.88.2...	8594919...	app.bilib...	157707339...	119.3.6...	app.bilib...
47	Social ...	10.89.7...	8e423dd...	cmshow...	157731748...	122.228...	cmshow...
62	News ...	10.88.1...	b86ea22...	ad-sdk...	157710886...	36.249...	ad-sdk...
86	Cateri...	10.89.2...	023000d...	mobile-p...	157706835...	39.98.2...	mobile-p...
103	Travel ...	10.89.7...	d4d3cd5...	push.12...	157707566...	39.98.7...	push.12...
215	Travel ...	10.95.2...	efd513e...	i0.dtslb....	157729092...	58.215...	i0.dtslb....
226	Search...	10.89.1...	9e3e398...	www.ba...	157753625...	180.101...	www.ba...
241	IT	10.89.1...	e27ae6f...	aod-ima...	157701451...	222.199...	aod-ima...

Figure 4. Sample data obtained after normalization

Among the columns, the ‘mysql_id’ field is a self-increasing field, which is not present in the original data. It was created to avoid the confusion of database sorting caused by the same ‘record_time’ field. The ‘category’ field is the category to which the record belongs. The ‘host_ip’ field is the user’s IP address. The ‘user_hash’ field is the hash value of the user ID that uniquely marks the user. The ‘dst_host’ field is the domain name of the visited site. The ‘record_time’ field is the access time. The ‘dst_ip’ field is the IP address of the site visited. The ‘url’ field is the record of the URL visited by the user.

3.1.2 Experimental Environment.

This experiment was executed in the following environment, CPU: Intel_Core_i3-9100F_3.60GHz; Memory: 16 GB DDR4 dual-channel RAM; Graphics card: NVIDIA RTX2060super 8 GB; Operating system: Windows 10 Professional Edition V1909.

3.1.3 Evaluation Criteria.

In this experiment, to evaluate the performance of the model effectively, the accuracy rate (p), mean absolute error (MAE), and root mean square error (RMSE) were used as the main evaluation criteria [12], which are defined by the following equations:

$$p = (TP)/(TP + FP) \quad (11)$$

$$E_{MAE} = \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i| \quad (12)$$

$$E_{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (|Y_i - \hat{Y}_i|)^2} \quad (13)$$

3.2. Experimental Results and Comparison with Different Models

The RSC user clustering model was constructed and implemented in Java. After repeated experiments, 24 user clusters were stably formed from 198 users. Compared with the K-means clustering algorithm, the RSC clustering algorithm can adaptively form multiple clusters, without any need to manually set the number of clusters. In this model, the RSC algorithm required less time and achieved better performance, and the clustering effect was clearly better than that of the k-means algorithm. Keras, which is a library for deep learning in Python, was used to build the proposed LSTM-attention model. To improve the efficiency of large-scale task training, adaptive moment estimation (Adam) [13] was used to optimize the training process, and the number of iterations was set to 100.

3.2.1. Prediction of URL Categories using the LSTM-Attention Model with Different Steps.

As the length of the input sequence of the LSTM-attention model tends to significantly affect the predicted results, input sequences with different sliding window step-sizes were used to find the best window step. Table 1 shows the results obtained after many

tests.

Table 1. Results obtained with different sliding window steps for the LSTM-attention model

Step	Accuracy	MAE	RMSE
1	0.1803	4.1647	8.9870
3	0.3363	3.9734	6.1037
7	0.3459	3.9293	7.3621
9	0.3391	4.0100	8.8222
11	0.2980	4.4423	9.8545

From Table 1, it can be observed that as the sliding window step increased, the accuracy of the LSTM-attention model initially improved and then degraded. The MAE and RMSE initially decreased and then increased. When the value of the sliding window step was 7, the LSTM-attention model achieved the best performance, and the accuracy for predicting the URL categories was 0.3459. Therefore, the sliding window step size was set to 7, and the predictive result of the URL category obtained with the LSTM-attention model is shown in Figure 5. (the predicted value is shown in yellow, while the true value is depicted in blue)

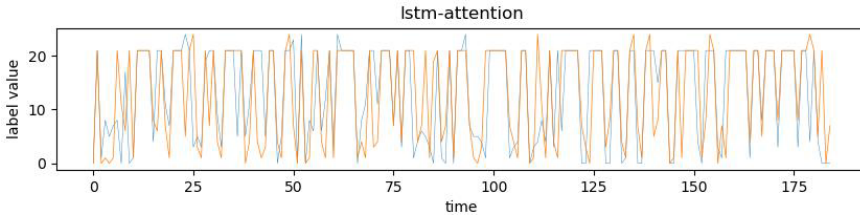


Figure 5. Results of LSTM-attention on the URL categories

(the predicted value of the model is shown in yellow; the true value is depicted in blue)

It can be concluded that with a small increase in the sliding window step, the model can fully learn the time information and the accuracy will increase accordingly. When the sliding window step exceeds 7 and becomes larger, the farther future needs to be predicted. As the model has more forward and backward propagation steps, the training time is longer, and the training curve clearly oscillates. Under such conditions, it will be considerably difficult for the model to predict the correct state of the system, leading to a decline in the performance.

3.2.2 Prediction Results of LSTM and LSTM-Attention on the URL Categories.

The predicted results of LSTM and LSTM-attention on the URL categories were compared to verify whether the attention mechanism improved the performance of the model. A comparison between the results of LSTM and LSTM-attention is shown in Table 2.

Table 2. Comparison between the results of LSTM and LSTM-attention on the URL categories

Method	Accuracy	MAE	RMSE
LSTM-Attention	0.3459	3.9293	7.3621
LSTM	0.2958	4.8764	9.8723

It can be observed that the predicted results of the LSTM and LSTM-attention models are similar to the URL category actually accessed by the users. Furthermore, the accuracy of the LSTM-attention model is slightly higher than that of the LSTM model. Therefore, the attention mechanism can effectively improve the model performance. In addition, the LSTM-attention model can reduce the network complexity and enhance

the generalization of the model.

3.2.3 Predicted Results of LSTM-Attention and Other Models on the URL Categories.

To verify the performance of the LSTM-attention model in predicting the URL categories, the LSTM-attention model was compared with the hidden Markov model (HMM) and BP neural network. The predicted results of different models on the same data set are shown in Figure 6, and the corresponding prediction errors are shown in Table 3.

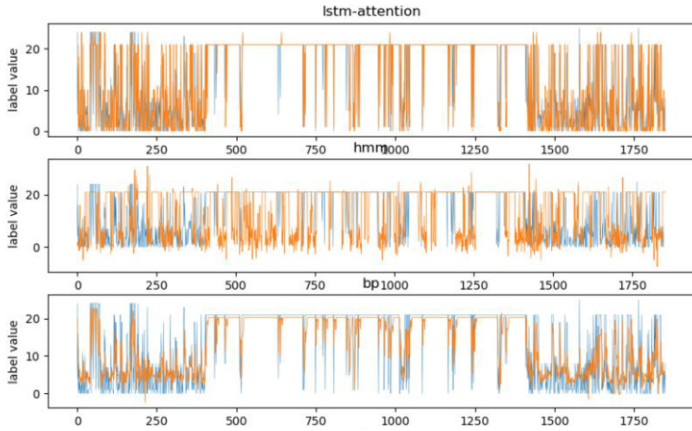


Figure 6. Results of LSTM-attention and other models on the URL categories (the predicted value of each model is shown in yellow; the true value is shown in blue)

Table 3. Comparison between LSTM and LSTM-attention on the URL categories

Method	Accuracy	MAE	RMSE
LSTM-attention	0.3459	3.9293	7.3621
HMM	0.2839	9.1528	12.4451
BP	0.2644	3.5057	5.6620

It can be observed from Figure 6 that, for the same data set, the prediction ability of the LSTM-attention model for the trend derived from the data is not as good as that of the BP neural network model. However, the accuracy of the LSTM-attention model is slightly higher than that of the HMM and BP neural network models, as shown in Table 3. The BP neural network has the highest predictive ability for the trend revealed from the data, with the trend with a small amplitude being the most sensitive.

3.2.4 Predicted Results of the LAR Model and Other Models on the Top-n URLs.

In the LSTM-attention model, the relationship between the URL category and frequency stems from a user's historical records. In the LAR model, the relationship between the URL category and frequency is established based on the group users' historical records generated after RSC clustering. The relationship between the URL category and frequency in the LSTM-attention + K-means model stems from the group users' historical records generated after K-means clustering. The accuracy of the top-n

URLs predicted by each model is shown in Table 4.

Table 4. Accuracy of the top-n URLs predicted by each model

Model	TOP 1	TOP 2	TOP 3	TOP 5	TOP 10
LSTM-attention	0.1010	0.1010	0.1263	0.2289	0.2310
LAR model	0.1010	0.1250	0.1466	0.2446	0.2501
LSTM-attention + K-means	0.1010	0.1105	0.1324	0.2295	0.2336

According to Table 4, when the value of N was 1, each model achieved a similar accuracy. When the value of N was 2, the accuracy of the LAR model was slightly higher than that of the LSTM-attention and LSTM-attention + K-means models. With increase in N, the performance of the proposed model gradually improved through using the historical records of the group users clustered by the RSC algorithm. It can thus be inferred that the RSC clustering algorithm significantly improved the accuracy of the model for URL prediction.

4. Conclusion

In this paper, a predictive model of user behavior from web log data based on the LSTM-attention network and RSC algorithm (called LAR) was proposed, which was used to classify and predict the web pages visited by users via the log data. In the proposed model, the LSTM-attention network was used to predict the category of the next URL that users might visit, and the RSC algorithm was used to cluster users. By using LSTM-attention to predict the URL category and the RSC algorithm to cluster users, combined with the URL category and URL mapping relationship, the prediction results of the LAR model provided the top-n URLs the users were likely to visit.

Through the experiments detailed in this paper, it can be concluded that the LSTM-attention network is slightly sensitive to different sliding window steps, and an improved performance is achieved when a moderate step value is used (in this study, the best value was determined to be 7). The accuracy of the predicted results for the URL categories by the LSTM-attention model was slightly higher than those of the HMM and BP neural network models. Furthermore, the RSC algorithm improved the predictive accuracy of the final URL. In summary, the LAR model is viable and effective for predicting a user's future access URL.

There are still some deficiencies in LAR model, which does not consider the influence of URL classification on the prediction results. In the future, we will consider using URL text as a direct input to the prediction model, eliminating aside the influence of classification. The predictive model proposed in this paper can be used for load balancing of servers or pre-caching of pages by browsers to achieve faster access. In addition, based on the predicted data, relevant data for public opinion surveys and early warning of abnormal and other applications can be obtained.

References

- [1] B.Harindra Varma, K. Ashok Reddy, Web User Behavior Analysis Using Improved Naïve Bayes Prediction Algorithm. *international journal of computer trends & technology* 5.2 (2013).
- [2] Dongshan X, Junyi S, A new Markov model for Web access prediction, *Computing in ence & engineering*, 4.6 (2002),34-39.
- [3] Awad M, Khalil I, Prediction of User's Web-Browsing Behavior: Application of Markov Model, *IEEE*

- transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society*, 42.4(2012), 1131-1142.
- [4] D Trincherro, R Stefanelli, L Cisoni, Integrating Markov Model with KNN Classification for Web Page Prediction, *International Journal of Computer Applications*, (2013).
 - [5] Panchal P S, Agravat U D, Hybrid technique for user's web page access prediction based on Markov model, *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, (2013).
 - [6] Pandey, Trilok, Ranjita et al, Merging Data Mining Techniques for Web Page Access Prediction: Integrating Markov Model with Clustering, *International Journal of Computer Science Issues*, (2012).
 - [7] Hochreiter S and Schmidhuber J, Long Short-Term Memory, *Neural computation*, 9(1997), 1735-1780.
 - [8] Bahdanau D, Cho K, Bengio Y, Neural machine translation by jointly learning to align and translate, *computer science*, (2014).
 - [9] Xie W B, Lee Y L, Wang C et al, Hierarchical Clustering Supported by Reciprocal Nearest Neighbors, *Information Sciences*, 7(2020), 279-292.
 - [10] Cho, Kyunghyun et al, On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, *Computer Science*, (2014).
 - [11] Greff K, Srivastava R K, Jan Koutnik et al, LSTM: A Search Space Odyssey, *Neural Networks and Learning Systems*, 28.10(2017), 2222-2232.
 - [12] C. J. Willmott, and K. Matsuura, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, *Climate Research* 30.1(2005),79-82.
 - [13] Kingma D, Ba J, Adam: A Method for Stochastic Optimization, *Computer Science*, (2014).