

## *Retraction*

# **Retracted: Prediction of Click-through Rate of Marketing Advertisements Using Deep Learning**

### **Wireless Communications and Mobile Computing**

Received 8 August 2023; Accepted 8 August 2023; Published 9 August 2023

Copyright © 2023 Wireless Communications and Mobile Computing. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

### **References**

- [1] M. Li, W. Sun, Q. Jia et al., "Prediction of Click-through Rate of Marketing Advertisements Using Deep Learning," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 1931965, 7 pages, 2022.

## Research Article

# Prediction of Click-through Rate of Marketing Advertisements Using Deep Learning

Mo Li,<sup>1</sup> WeiSheng Sun,<sup>1</sup> Qiaoran Jia,<sup>2</sup> Yilin Cui,<sup>1</sup> Shiru Li,<sup>2</sup> Liping Leah Wu,<sup>2</sup> Lixian Zheng ,<sup>1</sup> and Guanghui Qiao<sup>3</sup>

<sup>1</sup>City University of Macau School of Business Macau Special Administrative Region, 999078, China

<sup>2</sup>School of International Tourism and Management, City University of Macau Macau Special Administrative Region, 999078, China

<sup>3</sup>College of Tourism and Urban-Rural Planning, Zhejiang University of Technology and Industry, Hangzhou, Zhejiang, China 310000

Correspondence should be addressed to Lixian Zheng; 192911083@st.usst.edu.cn

Received 24 August 2022; Revised 14 September 2022; Accepted 20 September 2022; Published 15 October 2022

Academic Editor: Peiyong Zhang

Copyright © 2022 Mo Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the defect that the click-through rate of marketing advertisements cannot provide accurate prediction results for the company in time in the marketing strategy of Internet companies, this paper uses a deep learning algorithm to establish a prediction model for the click-through rate of marketing advertisements. The suggested model is called high-order cross-feature network (HCN). Furthermore, this paper also introduces the combination of feature vectors into the graph structure and as the nodes in the graph; therefore, the graph neural network (GNN) is used to obtain the high-level representation ability of structured data more fully. Through numerical simulations, we observed that HCN has the capability to provide Internet companies with more accurate advertising business information, user information, and advertising content. Moreover, HCN model is more reasonable to adjust the advertising strategy and can provide better user experience. The simulation outcomes indicate that the suggested HCN approach has noble adaptability and high correctness in forecasting the click-through rate of marketing advertisements. We observed that this improvement, in terms of predictions precisions and accuracies, can be as high as 17.52% higher than the deep neural network (DNN) method and 10.45% higher than the factorization network (FM) approach.

## 1. Introduction

In traditional applications, the historical click-through rate of advertisements is usually counted as a prediction result, and then the advertisements are sorted according to the historical click-through rate. However, the problem with this method is that the statistical click-through rate is only a fixed value, and it is difficult to carry out personalized advertising according to the preferences of specific users [1]. In fact, the difference between advertising space, users and advertising content will affect the probability of users clicking on the advertisement. In recent years, the development of machine learning algorithms has led researchers to consider the problem of advertising click-through rate estimation as a typical two-class problem, that is, to directly

predict the probability that an upcoming advertisement will be clicked by the current user.

Aoying et al. [2] proposed a new framework to train logistic regression models in parallel, and experiments showed that using this framework can reduce the training time by an order of magnitude. Although traditional linear models have application advantages, these models lack the ability to learn cross-features, and feature engineering of data often directly affects the performance of the model. Therefore, in order to improve the effect of the model, it is necessary to spend a lot of time artificially constructing combined features for better performance. However, even experienced data experts cannot construct all the hidden feature information. Therefore, the authors in [3] proposed to learn the original features through GBDT, and then add

the number of the leaf nodes of each tree as new features to the logistic regression model. The suggested approach has the capability to solve the feature combination problem of linear models. Of course, in real scenarios, the CTR estimation is not a simple linear problem. Therefore, some researchers have begun to consider how to increase the nonlinear relationship to solve the CTR estimation problem. Agarwal et al. [4] used the hybrid logistic regression algorithm to directly introduce piecewise linearity in the original data space to fit the high-dimensional nonlinear data distribution. And it has end-to-end learning ability, which saves a lot of artificial feature design.

Scholars believe that different keywords have different probability of being clicked by users. Therefore, by analogy to advertisements, the distribution of click-through rate also follows this principle. Therefore, the clustering algorithm is used to cluster keywords to obtain new information, so as to improve the accuracy of the model when the user interaction data is insufficient. Gai et al. [5] used the relationship between search term and advertisement pair to build a user click association graph and extracted the statistical information related to the search term or advertisement from the perspective of the relation graph. However, it is very inappropriate to consider all feature intersections, and not all feature combinations will bring benefits, so they use decision trees to train the model and use the feature importance output of the decision tree to selectively prune features to remove redundancy. References [6–8] improve the prediction performance of the model. Online advertising display is a very promising field. How to display the right advertisement to the customer at the right time and place is a core algorithm problem of the advertisement display system. Therefore, accurate prediction of the click-through rate of advertisements has become one of the most important technologies in advertising algorithms [9, 10]. The following are the major contributions of our research:

- (i) We use a deep learning algorithm to establish a prediction model for the click-through rate of marketing advertisements
- (ii) The suggested model HCN also introduces the combination of feature vectors into the graph structure, and the graph neural network (GNN) is used to obtain the high-level representation ability of structured data more fully
- (iii) The simulation outcomes indicate that the suggested HCN approach has noble adaptability and high correctness in forecasting the click-through rate of marketing advertisements

The rest of the paper is structured as follows: the higher-order cross-feature depth model (HCN) is deliberated in the next Section 2. In Section 3, dataset and training of deep models are discussed in more detail. Numerical experiments, validations, and obtained results are discussed in Section 4. Finally, we summarize the paper in Section 5 and discuss some future insights that can be used by the researcher to take our work into the next levels.

## 2. Higher-Order Cross-Feature Depth Model

**2.1. High-Order Cross-Feature Network.** For a sample, since each domain generally has one and only sparse feature value of 1, after the embedding layer index is taken, a domain has only one embedding vector  $v_{1,i}$ , and  $i$  is the domain index. Then, the cross-feature vector of the first layer is mathematically expressed.

$$Z_1^{i,j} = NDP(v_{1,i} \cdot v_{1,j})w_1^{i,j}. \quad (1)$$

In Equation (1),  $Z_1^{i,j}$  is the cross eigenvector of the eigenvectors  $v_{1,i}$  and  $v_{1,j}$  of domain  $i$  and domain  $j$  in the first layer. Similarly, NDP is the dot product normalization operation, and  $w_1^{i,j}$  represents the vector-level weight for domain combination.

For the obtained cross-feature vector, the vector representation of the new sparse feature can be reconstituted according to the domain belonging, and then the calculation formula of each domain feature vector of the  $l^{\text{th}}$  layer is expressed using the following:

$$v_{l,t} = \sum_{i=1}^n \sum_{j=i+1}^n |t \in \{i, j\}| Z_{l-1}^{i,j} \cdot h_1^{i,j} + v_{l,t}. \quad (2)$$

In Equation (2),  $t$  represents the index of the domain,  $n$  is the number of domains, and  $v_{l,t}$  represents the newly reconstituted sparse feature vector representation in the domain numbered  $t$  in the  $l^{\text{th}}$  layer. This should be noted that  $t \in \{i, j\}$  means that if  $t$  is equal to  $i$  or  $j$ , then the value is 1, otherwise the value is 0. Furthermore,  $h_1^{i,j}$  is the weight vector to be learned. The vector  $v_{l,t}$  reweights and sums all the cross-feature vectors related to the domain  $t$  in the previous layer to reconstruct the new sparse feature vector representation and introduces the embedding vector  $v_{l,t}$  to avoid the loss of the underlying information. In the next step, we recalculate the cross eigenvectors on the newly obtained vector representation, then the cross-feature vector of the first layer, as stated in equation, is recomputed using the following:

$$Z_l^{i,j} = NDP(v_{l,i} \cdot v_{l,j})w_l^{i,j}. \quad (3)$$

Similarly, next we iteratively calculate the above three formulas, i.e., Equations (1)–(3), in order to obtain the cross-feature information of different orders. In order to preserve the cross information of different levels as much as possible, the HCN sums the cross-feature vectors of each layer and outputs as

$$X_l^{\text{out}} = \sum_{i=1}^n \sum_{j=i+1}^n \sum_{r=1}^k z_l^{i,j,r}, \quad (4)$$

where  $k$  is the dimension size of the embedding vector. If the Sigmoid activation function is used after summing the  $X_l^{\text{out}}$  of all layers, then it can be used as a separate high-order

cross-feature model HCN, and the structure is shown in Figure 1. If the linear module and the depth module are combined, then the calculation formula of the DCFM model is illustrated as

$$\hat{y}_{\text{DCFm}} = \sigma \left( w_0 + \sum_{i=1}^m x_i w_i + \sum_{l=1}^L X_l^{\text{out}} + \text{MLP}(v_{\text{concat}}) \right), \quad (5)$$

where MLP is the output of the depth module, and  $L$  is the number of HCN layers. The HCN model continuously reconstructs new feature vector representations for sparse features, and then uses crossover operations to form higher-order cross-feature information. The entire process of the HCN model is shown in Figure 1.

**2.2. The Choice of Learning Rate Method.** In the advertisement click-through rate estimation scenario, the data set is very huge, often millions or even tens of millions. If the traditional fixed learning rate method is used, training will be very time-consuming. At the same time, if the learning rate is set too large, the training time will be shortened, but it may oscillate around the minimum value and fail to converge, and the accuracy will be greatly reduced. If the learning rate is set too small, although the accuracy will be improved, it will be very time-consuming. In order to find a perfect balance between training time and accuracy, this article will use a degenerate learning rate, also known as learning rate decay. The degenerate learning rate combines the advantages of two methods of large learning rate and small learning rate in the training process, that is, using a large learning rate at the beginning of training to speed up training. After training to a certain extent, reduce the learning rate to improve the accuracy and reach the convergence state. The calculation expression is mathematically expressed and given in the following:

$$\text{learning\_rate} = \text{learning\_rate\_base} * \text{learning\_rate\_decay} * \frac{\text{global\_step}}{\text{decay\_step}}. \quad (6)$$

In Equation (6), `learning_rate_base` represents the initial value of the learning rate. `learning_rate_decay` represents the learning rate decay coefficient, and `global_step` represents the number of rounds of sample training. `Decay_step` represents the step of decay. For example, set `learning_rate_base` to 0.1, `learning_rate_decay` to 0.9, `global_step` to 50, and `decay_step` to 10. Then the `learning_rate` is 0.1 at the beginning of the training, the `learning_rate` is 0.09 when the training reaches the 20th round, and so on, and the `learning_rate` is 0.081 at the 40th round, as made known in Table 1.

This can be easily understood and comprehended from the values given in Table 1 that the learning rate changes relatively large at the beginning, and the later changes are small, which can meet the needs of a large learning rate in the early stage to speed up the training speed, and a small learning rate in the later stage to improve the accuracy and achieve the purpose of convergence.

**2.3. Selection of Activation Function.** In order to avoid gradient dispersion and speed up the training speed, the ReLU activation function is generally used when selecting the activation function. The function image of the ReLU activation is shown in Figure 2.

It can be seen from Figure 2 that the ReLU activation function outputs all values less than 0 as 0. The derivative of some functions greater than 0 is a fixed value, so when backpropagating, the calculation is simple, and the training speed can be accelerated. However, since all values less than 0 are replaced with 0, therefore there is a great possibility that all data will be 0, and the model cannot continue during the backpropagation process. In order to speed up the training speed without discarding all negative values in the ReLU, this article will use the activation function which is known as the LeakyReLU and is a variant of the classical ReLU activation. The image of the LeakyReLU function is shown in Figure 3.

### 3. Training of Deep Models

**3.1. Dataset Information.** This paper tests the model effect on three general public datasets. The specific data set statistics are shown in Table 2. The Criteo dataset is a dataset released by Criteo, a world-renowned advertising company, during a display advertising competition. The dataset comes from the logs recorded by users when they visit web pages and contains 40 million data samples, each of which has 13 numerical features and 26 discrete features. All features are anonymous, so there is no way to know the specific meaning of the features. The Conversion dataset contains historical conversion feedback for over 1 million clicked ads over a 2-month period. Each row of samples represents a historically clicked display advertisement information, which consists of 8 numerical features and 9 discrete features. The discrete features are hashed into 32-bit integer numbers for the purpose of protecting user privacy, and some features may have missing values. To reduce the size of the original data, the dataset has been down sampled. Similarly, the Avazu dataset is a dataset for real-time advertising algorithm competition shared by a foreign advertising company known as Avazu in 2015. In fact, the dataset records information such as the time, advertisement position, URL, and device type when an advertisement is displayed within 10 days. All features of the sample are discrete features and have been desensitized.

**3.2. Model Parameter Settings.** For dataset processing, refer to the process of processing Criteo datasets in the open source framework Paddle. For discrete features, after counting the number of occurrences in the entire data set, the features with less than a certain number of occurrences are attributed to one feature; for numerical features, the features with more than 95% quantile values are directly assigned to 95% after ascending order. The corresponding value is located at the quantile of the dataset. Each feature is then mapped to an integer index value using a label encoding algorithm. In the model Embedding layer, the embedding vector corresponding to the sparse feature is found directly

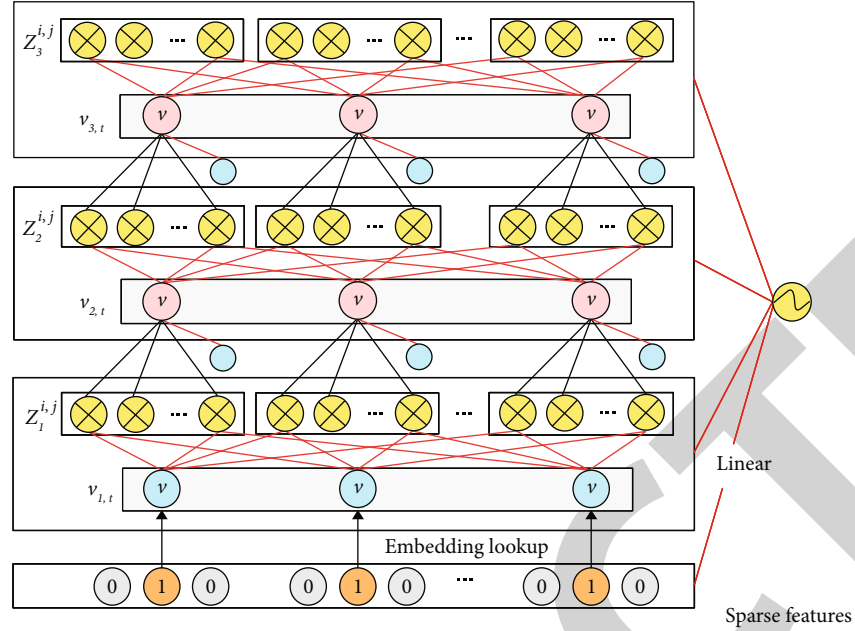


FIGURE 1: The three-layer HCN model structure.

TABLE 1: The learning rate attenuation.

Number of training rounds	Attenuation rate	Learning rate
0	0.9	0.12
10	0.9	0.08
20	0.9	0.086
30	0.9	0.071
40	0.9	0.062
50	0.9	0.058

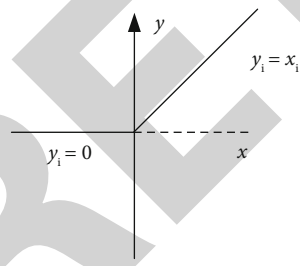


FIGURE 2: The ReLU activate function.

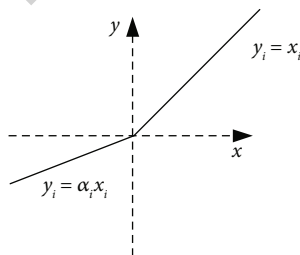


FIGURE 3: The LeakyReLU activation function.

through the index value. The training set, validation set, and test set are divided into 60%, 20%, and 20% in a chronological order. All models use Adam algorithm as parameter optimizer. The model parameters are determined using a grid search algorithm. The range of the learning rate is mainly [0.001, 0.0005, 0.0001, 0.00005, 0.00001], the range of the L2 penalty coefficient is mainly [0.0001, 0.00001, 0.000001, 0.0000001], the dimension of the embedding vector is set to 50 by default unless otherwise specified, and the batch size during training, for 2048, individual models may require finer-grained parameter tuning. The model adopts the early stopping method, and the model training will be terminated when the model has not significantly improved on the validation set after 3 iterations.

#### 4. Analysis of Experimental Results

In order to authenticate the prediction ability and correctness of the HCN module, the HCN module is split out for experiments. The results are shown in Table 3. Among them, the DNN means that only the deep model is used, and Cross network means that only the cross-feature part in the DCN model is used. In order to make up for the lack of linear features, the Cross network and HCN models here also include linear LR modules. The value after the symbol “l” indicates the number of layers used. It can be seen that when the number of layers of HCN is 1, then the model structure is only one more dot product normalization operation than FvFM, but its effect is still a good improvement. Since, the effect of numerical magnitude recovery is not very large when the number of layers is low, this may be mainly due to the nonlinear factors brought by dot product normalization. When taking a better number of layers, compared with the Cross network in the DCN model, the HCN model has achieved greater advantages. This is evident from the

TABLE 2: Detailed statistics of each discrete dataset.

Data set	Number of samples	Number of numeric fields	Number of discrete domains	Number of features after processing
Criteo	44640677	14	28	121632
Conversion	16764869	12	9	33651
Avazu	41448978	0	24	112895

TABLE 3: High-order cross-model HCN experimental results AUC.

Model	Criteo	Conversion	Avazu
FvFM	0.7684 1	0.8418 1	0.7442 1
Cross network	0.7915 3	0.8415 3	0.7376 3
DNN	0.8124 3	0.8442 3	0.7468 3
HCN	0.8156 1	0.8436 1	0.7452 1
HCN	0.8094 3	0.8445 2	0.7486 3

accuracy of the HCN and other model as discussed later in this section.

The feature crossover mechanism of the HCN model still retains the domain relationship, and the crossover operation is more in line with the logical “and” thinking, unlike the Cross network model, it is only a matrix mapping of the overall splicing features, and only the highest-level feature information is output. Compared with the DNN model, the model effect of the HCN model is not too bad, and there are advantages and disadvantages in the data set. Because these two modules use different forms of high-level feature information, they have both certain accuracy and certain differences in the model effect. Therefore, when the two modules are jointly trained, the feature information of different perspectives is complementary, and the model is not affected. The improvement of decision-making is very obvious, and the mechanism of action is similar to ensemble learning in machine learning.

In order to examine and evaluate the influence of the numeral quantity of layers used by the HCN on the model effect, experiments are carried out on different layers, and the consequences are displayed in Figure 4. Through investigating these values, this can be found that when the numeral amount of layers is from 1 to 2, then the AUC of the model is improved significantly. At this time, the crossover order equivalent to the feature has been raised from order 2 to order 3-4. However, when the number of layers exceeds 3, then the effect of the model does not decrease, but there is basically no upward trend. This also means that a too high crossover order will not produce large fluctuations in the model performance. Similarly, when the number of domains is quite and potentially large, then the number of parameters reduced by using a smaller number of layers is still considerable.

Figure 4 compares the AUC variation curves of the models under different embedding vector dimensions. Observing the trend of the curve, the factorization model (FM) is more sensitive to the value of the embedded vector

dimension, and the model effect increases with the increase of the embedded vector dimension, until the value of 25 or later, the improvement is no longer obvious. On the contrary, the DNN-based models DeepFM and DCFM do not have high requirements on the dimension of the embedding vector. It can be seen that the embedding vector of different sizes does not have a great impact on the model. This may be due to the fact that the factorization class model belongs to the shallow model and needs more parameters to represent the semantics of the sparse features, while the deep neural network has a deeper network structure and can still learn high-level from the embedding vector of smaller dimensions and semantic features.

For both datasets, Figures 5 and 6 illustrates the accuracy of the various approaches against the suggest HCN model. For both datasets, we observed that the HCN model is significantly more accurate than the other approaches. The RMSE and MAPE values also demonstrate the supremacy of the HCN model.

## 5. Conclusions and Future Work

Faced with the problem of insufficient research on constructing high-order cross eigenvalues, this paper solves this problem by retaining the underlying FM structure and recombining cross eigenvalues into new sparse eigenvectors. Then repeat the steps of cross sparse eigenvectors of eigenvalues to obtain cross eigenvalue information of different orders. In order to avoid the problem that the magnitude of the output value is too small caused by multiple vector dot products, a dot product normalization operation is designed to effectively carry out gradient backpropagation. And the linear module and the depth module are combined to form an end-to-end learning of different levels of feature information. The effects are fully evaluated and compared on three public datasets, and experiments show that the algorithm has certain advantages over other prediction models. The simulation outcomes indicate that the suggested HCN approach has noble adaptability and high correctness in forecasting the click-through rate of marketing advertisements. We observed that this improvement, in terms of predictions precisions and accuracies, can be as high as 17.52% higher than the deep neural network (DNN) method and 10.45% higher than the factorization network (FM) approach.

In the future, advanced deep learning techniques like deep neural networks (DNNs) must be taken into account to boost the exactness of the prediction conclusions. The time-consuming training procedure for learning algorithms has the potential to reduce the system’s overall performance. We will, in the near future, contemplate separating the training and prediction parts across the edge-cloud structural design for the purpose that the training could take place at the distant cloud, which normally has a lot of computational assets and power. Contrary to this, the prediction fragment of the procedure would execute on the cutting edge that, subsequently, will significantly shorten the system’s response and processing or execution times. Furthermore, more

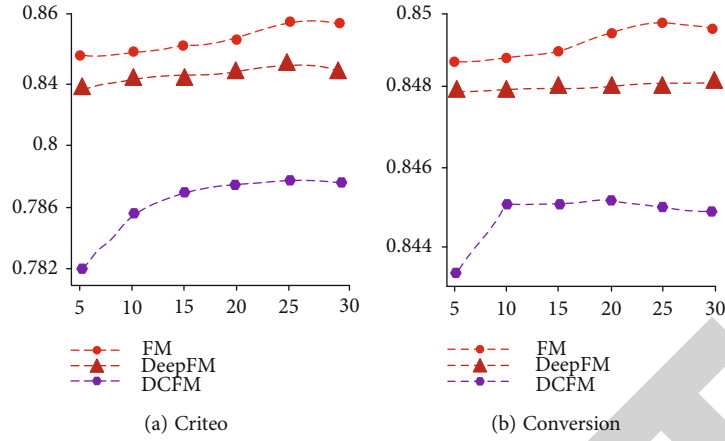


FIGURE 4: The effect of different embedding vector dimensions on the model.

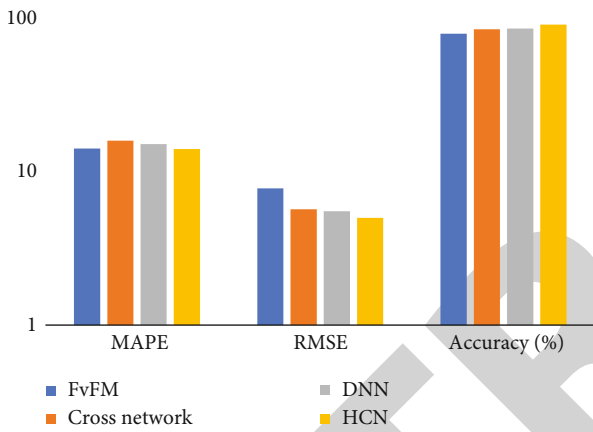


FIGURE 5: Comparison of various approaches in terms of accuracy (Criteo).

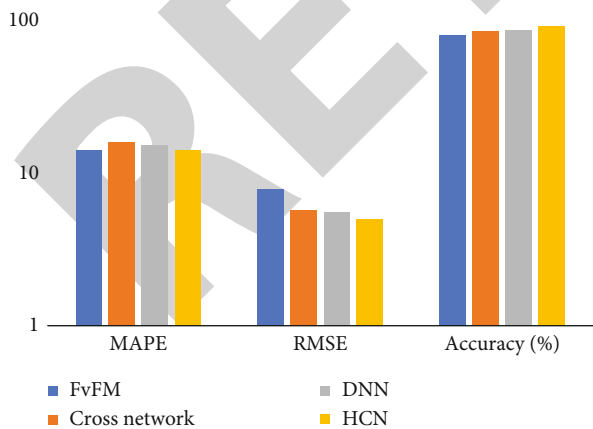


FIGURE 6: Comparison of various approaches in terms of accuracy (Conversion).

robust and faster deep learning approaches will be developed to improve the accuracy.

**Data Availability**

The data used to support the findings of this study are available from the corresponding author upon request.

**Conflicts of Interest**

The authors declare that they have no conflict of interest.

**Authors' Contributions**

Mo Li and WeiSheng Sun have contributed equally to this work and share first authorship.

**References**

- [1] H. B. McMahan, G. Holt, D. Sculley et al., "Ad click prediction: a view from the trenches [C]," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, United States, 2013.
- [2] Z. Aoying, Z. Minqi, and G. Xueqing, "Computational advertising: web comprehensive application with data as the core [J]," *Chinese Journal of Computers*, vol. 34, no. 10, pp. 1805–1819, 2011.
- [3] Z. Zhiqing, Z. Yong, and X. Xiaoqin, "Research on advertising click-through rate prediction technology based on feature learning [J]," *Chinese Journal of Computer*, vol. 39, no. 4, pp. 780–794, 2016.
- [4] A. Agarwal, O. Chapelle, M. Dudík, and J. Langford, "A reliable effective terascale linear learning system [J]," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1111–1133, 2014.
- [5] K. Gai, X. Zhu, H. Li, K. Liu, and Z. Wang, "Learning piecewise linear models from large scale data for ad click prediction [J]," 2017, arXiv preprint arXiv:1704.05194.
- [6] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*, pp. 995–1000, Sydney, NSW, Australia, 2010.

- [7] Y. Juan, Y. Zhuang, W. S. Chin, and C. J. Lin, "Field-aware factorization machines for CTR prediction," in *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 43–50, New York, NY, United States, 2016.
- [8] W. Zhang, T. Du, and J. Wang, "Deep learning over multi-field categorical data," in *European Conference on Information Retrieval*, pp. 45–57, Springer, Cham, 2016.
- [9] Y. Qu, H. Cai, K. Ren et al., "Product-based neural networks for user response prediction," in *2016 IEEE 16th ICDM*, pp. 1149–1154, Barcelona, Spain, 2016.
- [10] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD*, p. 12, New York, NY, United States, 2017.

RETRACTED