IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 7, 2021, accepted February 23, 2021, date of publication February 26, 2021, date of current version March 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3062687

Research on Disease Prediction Based on Improved DeepFM and IoMT

ZENGCHEN YU[1], SYED UMAR AMIN[2], MUSAED ALHUSSEIN[2], AND ZHIHAN LV[1], (Senior Member, IEEE)

[1]School of Data Science and Software Engineering, Qingdao University, Qingdao 266071, China
[2]Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding author: Syed Umar Amin (samin@ksu.edu.sa)

This work was supported by the Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia, through the Research Group Project, under Grant RG-1436-016.

ABSTRACT In recent years, with the increase of computer computing power, Deep Learning has begun to be favored. Its learning of non-linear feature combinations has played a role that traditional machine learning cannot reach in almost every field. The application of Deep Learning has also driven the advancement of Factorization Machine (FM) in the field of recommendation systems, because Deep Learning and FM can learn high-order and low-order features combinations respectively, and FM's hidden vector system enables it to learn information from sparse data. The integration of them has attracted the attention of many scholars. They have researched many classic models such as Factorization-supported Neural Network (FNN), Product-based Neural Networks (PNN), Inner PNN (IPNN), Wide&Deep, Deep&Cross, DeepFM, etc. for the Click-Through-Rate (CTR) problem, and their performance is getting better and better. This kind of model is also suitable for agriculture, meteorology, disease prediction and other fields due to the above advantages. Based on the DeepFM model, we predicts the incidence of hepatitis in each sample in the structured disease prediction data of the 2020 Artificial Intelligence Challenge Preliminary Competition, and make minor improvements and parameter adjustments to DeepFM. Compared with other models, the improved DeepFM has excellent performance in AUC. This research can be applied to electronic medical records to reduce the workload of doctors and make doctors focus on the samples with higher predicted incidence rates. For some changing data, such as blood pressure, height, weight, cholesterol, etc., we can introduce the Internet of Medical Things (IoMT). IoMT's sensors can be used to conduct transmission to ensure that the disease can be predicted in time, just in case. After joining IoMT, a healthcare system is formed, which is superior in forecasting and time performance.

INDEX TERMS Deep learning, factorization machine, hepatitis, Internet of Medical Things.

I. INTRODUCTION

In the 1950s and 1960s, neural networks were proposed. Inspired by human brain neurons, this model consists of multiple neurons that receive and process signals from connected neurons/nodes. Each neuron receives one or more inputs to change its internal state (activation) [1]. The learning process of this model is: each input is multiplied by a weight and transmitted to the neuron. After that, the current neuron calculates the sum of all the inputs including the bias values, and through the activation function for nonlinear conversion and output. Follow these steps to the next neuron [2]. After completing the forward propagation step above, then back propagation [3], calculate the error of each weight parameter and subtract the corresponding error, repeat these steps several times to learn. This structure is shown in Fig. 1, including an input layer, one or more hidden layers, and an output layer [4]. The input layer receives input features, then the hidden layers perform a series of nonlinear transformations, and the output layer outputs the final result.

The multi-layer neural network can learn any function with arbitrary precision. But the computing power of computers at that time was not powerful, and the Graphics Processing Unit (GPU) did not appear. Until recent years, after the hardware was satisfied, Deep Neural Networks (DNN), or Deep Learning, began to take the stage due to its powerful

The associate editor coordinating the review of this manuscript and approving it for publication was Abdulsalam Yassine.

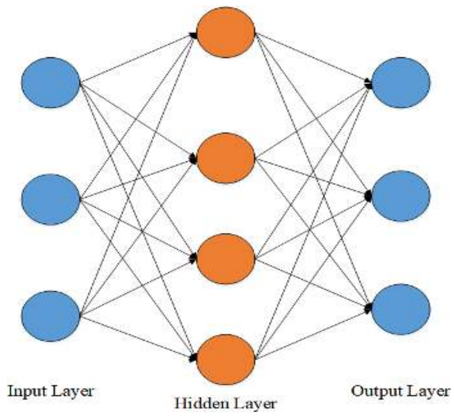


FIGURE 1. Neural Network diagram.

learning capabilities. Deep Learning has good adaptability to structured and unstructured data. Therefore, many scholars aim at this feature and apply it to recommendation system [5], medicine, and other aspects.

With the application and popularization of Deep Learning, Factorization Machine (FM) has also been developed. FM was first proposed by Rendle in 2010. This is a supervised learning model [6], which combines the advantages of matrix factorization and support vector machines (SVM) [7]. It is similar to learning methods such as Logistic Regression and SVM [8]. The difference is that FM uses decomposition interactions between hidden variables, which can better mine the feature combination information of second order or above, and can achieve better results in the case of sparse data [9], [10]. FM was first applied to Click-Through-Rate (CTR) prediction in the recommendation system [56] to mine the combination of features behind users' click behavior. Many scholars have made many improvements based on FM. For example, Field-aware Factorization Machine (FFM) introduced the concept of field, that the features with the same features are grouped into a field, and each field's feature interact with those of other fields to varying degrees., These field information will be fully utilized [11], but the model complexity is $O(kn^2+kn)$, k is the latent vector length, n is the number of features; Field-weighted Factorization Machine (FwFM) solves the problem of parameter explosion caused by the multiplication of the number of parameters, features, and fields in FFM, which introduces the interactive strength weight of the field and learns the hidden vector [12]. The above two are applicable to the situation where there are too many attributes and their eigenvalues. For the hierarchical relationship between context features, such as inclusion and dependence between attributes, User Weight Factorization Machine (UW-FM), Item Weight Factorization Machine (IW-FM) model, Category Weight Factorization Machine (CW-FM) model, Hierarchical Importance-aware Factorization Machine (HIFM), etc. can play a role. For the case where the attributes are independent of each other, it is better to use the basic FM to avoid the model being too complicated. The data used in this article are basically independent of each other, so FM is used as the model of low-level feature combination.

In many cases, FM only involves second-order feature interaction, which is a shallow model, but in real life data is often highly nonlinear, as is the case with recommendation systems. High-order feature interaction is essential for good performance [13]. Although theoretically FM can fit high-order feature combinations, such calculations are too large, and time complexity and storage space consumption will explode. If the feature combination is performed manually, there are the following shortcomings: first, it requires experts in related fields to spend a lot of time to study the relationship of related features, which is time-consuming and laborious, especially when the number of features is more than several hundred. Second, for large-scale prediction systems, the amount of data is huge, manual feature extraction is unrealistic. Third, it is impossible to generalize features interaction that is not in training [14]. Therefore, Deep Learning can be added for automatic learning of high-level features combinations, which is one of its advantages over traditional machine learning. Correspondingly, there are various researches that combine FM with deep learning models, such as Factorization-supported Neural Network (FNN), Product-based Neural Network (PNN), Wide&Deep, Deep&Cross, DeepFM, etc., which will be introduced in detail below.

However, in practical applications, some features are not static. Collecting these data in a timely manner and re-forecasting can reduce the occurrence of risks. With the remarkable advancement in smart sensors, cloud computing technologies and internet of things (IoT), the concept of smart healthcare has revolutionized the entire healthcare industry. To tackle massive amount of real time health data, IoT has given viable solutions. The convergence of IoT with Artificial Intelligence (AI) has transformed the healthcare industry by employing state of the art techniques like Deep Learning, cloud, fog and edge computing etc. The massive amount of medical data can now be leveraged using advanced AI techniques to provide intelligent, accurate and real-time healthcare services. AI together with IoT has also provided great smart personalized healthcare solutions based on the deep insights of the medical data. IoT technologies has also provided huge storage and powerful computing capabilities for processing big data.

The concept of the Internet of Medical Things (IoMT) needs to be introduced here. The Internet of Things is a technology mode for quickly acquiring remote information through modern wireless communication technology [15], which aims to "connect all things", that is, in an environment, all people and things, things and things are connected through the network. This technology is based on the information collected by ubiquitous devices such as sensors, software platforms, and mobile phones to achieve communication and collaboration between objects [16]. Due to the above characteristics, the Internet of Things can be applied to medicine, that is, IoMT, which monitors the health of users by tracking the health records of patients [17], captures abnormal situations, and reminds users of their own health risks. Wearable sensors are now generally used to collect specific data from the body [18], such as blood sugar levels, exercise, weight, blood pressure, etc., and then transmitted to the cloud for cloud computing, or edge computing locally or fog computing on a virtual platform in order to realize the application. Edge computing has also helped realize cognitive IoT frameworks [19]. In recent years, the application of IoMT has been on the rise [20]. The combination of Deep Learning and IoMT has become one of the current hot spots. Many scholars have conducted research on the combination of the two and have obtained many results.

We will use DeepFM to predict the incidence of hepatitis on structured disease data, and make micro-improvements and parameter adjustments to make it more in line with the actual learning situation, and compare it with Logistic Regression, DNN, FNN and PNN algorithms in Area Under the Curve (AUC) and Logistic Loss evaluation indicators. In addition, a timely prediction scheme of Internet of Things combined with edge computing is proposed, so as to realize the combination of DeepFM and IoMT and form a basic healthcare system.

This article will make the following arrangements: Section 2 will briefly introduce the application of Deep Learning and the IoT and FM in diseases, and list the development history of the integration of Deep Learning models and FM models, thus leading to the conclusion that the combination of Deep Learning, IoT and FM has a substantial effect on the medical industry; Section 3 introduces and explains the DeepFM model and the improvement of the weight hyper-parameters in this article, and the combination with IoMT; Section 4 explains the data used, evaluation indicators, parameter optimization and experiment results; Section 5 draws conclusions.

II. RELATED WORKS

A. THE APPLICATION OF DEEP LEARNING AND THE IOT IN HEALTHCARE AND MEDICINE

Deep Neural Networks does have a good application in medicine and smart healthcare. It has good adaptability in the analysis of multimodal healthcare data (image, text, speech, structured data) [21], [60], [62] in a smart connected healthcare domain [61]. Sumit Sharma *et al.* have proposed Recurrent Neural Network (RNN)-based Alzheimer's prediction scheme that uses trigger-based sensors to collect sensorimotor data in the Internet of Health (IoH) [57], [58] ecosystem, which is nearly 10-20% more accurate than existing machine learning algorithms [22]. Hossain proposed a cloud-based patent and health monitoring model in the cyber physical environment, which has high efficiency and accuracy [23]; Deep Convolutional Neural Network (DCNN) and IoT are used in oral cancer image classification. It can achieve 96.8% accuracy and 92% sensitivity [24]. In order to protect patient privacy, Xiang Wu *et al.* proposed a medical big data privacy protection platform based on the IoT to facilitate the safe sharing of medical data [25]. Therefore, it is feasible to apply Deep Learning and the IoT to medicine.

B. DEVELOPMENT OF FM AND DEEP LEARNING INTEGRATED MODEL

In 2016, Zhang *et al.* proposed FNN. This model performs FM preprocessing and initialization of the original features in the embedding layer, and then inputs the DNN for learning, so it is limited by the ability of FM [26]. In the same year, Qu *et al.* introduced the product layer between the embedding layer and the full connection layer, that is, adding the function of pair crossing when embedding features, and proposed PNN, which helps to highlight the interaction between different features, but may ignore the valuable information contained in the original vectors [27]. However, the former two have a common shortcoming: low-level feature interactions cannot make reasonable contributions to the model. Only high-level feature interactions will be transmitted to the output layer [26]. But it is also necessary to capture low-level feature interactions information. That broad and deep learning is better at solving regression and classification problems. It integrates wide learning units and deep learning units. The wide learning units are generalized linear models composed of single-layer perceptions. The deep learning units are multi-layer perceptions, which are respectively responsible for learning from historical data and creating abstract representations [28].So Google proposed the Wide&Deep model in 2016, which is divided into two parts: width and deep part. Width part is the width model, based on Generalized Linear Models, and the mathematical formula is:

$$y_{wide} = \boldsymbol{w}^T \boldsymbol{x} + b \tag{1}$$

Among them, $\boldsymbol{x}$ is the feature vector, $\boldsymbol{w}$ is the weight matrix, b is the bias term, $\boldsymbol{x}$ includes the original features and some feature interactions selected by humans; the depth part uses DNN, which focuses on automatically constructing features. The mathematical formula is:

$$\boldsymbol{a}^{(l+1)} = \sigma(\boldsymbol{W}^{(l)}\boldsymbol{a}^{(l)} + \boldsymbol{b}^{(l)}) \tag{2}$$

l is the number of layers, σ is the activation function, $\boldsymbol{a}^{(l)}$, $\boldsymbol{W}^{(l)}$, $\boldsymbol{b}^{(l)}$ are the input vector, weight parameter matrix, and deviation of the layer l, respectively.

However, the features of the width part need to be manually designed because it will directly participate in the prediction. If the structure of this part is not accurate, it will affect the accuracy of the entire model [29]. Deep&Cross and Wide&Deep share a similar framework [30], but it uses a residual network instead of a width model to obtain low-level interactive information. But whether it is Wide&Deep or Deep&Cross, the width and depth parts require different inputs. Neural Factorization Machine (NFM) uses a framework similar to Wide&Deep. It seamlessly combines the advantages of neural networks and factorization machines. It not only depicts the linear interaction between features, but also simulates the nonlinear high-order feature interaction between them [31], but the second-order feature interaction of this model is used as the input of the depth part. DeepFM is similar to NFM in that its width and depth parts share the same input, and the first-order features and the interaction of second-order and high-order features will be simultaneously input to the output layer. Pre-training and feature engineering are not required [32]. Experiments show that the performance of DeepFM is better than those of the models mentioned above.

C. MEDICAL APPLICATIONS OF FM MODELS

FM, the integrated model of FM and Deep Learning are mainly used for recommendation systems, especially CTR problems. Many features in this type of data are discrete [29], and are very sparse after one-hot encoding. Other fields also have the same features, so many scholars have applied FM, DeepFM, etc. to other fields such as agriculture, meteorology [33], and medicine, and achieved good results. FM and its derivative algorithms are rarely studied in the medical field, but there are a few research results.

Makoto Yamada *et al.* proposed Convex Factorization Machine (CFM) model based on FM to predict drug toxicity. This problem is optimized as a semi-definite programming problem to find the global optimal solution [34]; With Deep Learning models, it's going to work really well. Yanghua Fan *et al.* applied DeepFM to predict the recurrence of Cushing's disease after transsphenoidal surgery, and used 354 patients in Peking Union Medical College Hospital to predict recurrence and obtained the highest AUC value (0.869). And the lowest Logistic Loss value (0.256), which exceeds models such as Multilayer Perception (MLP), Logistic Regression (LR), Gradient Boosting Decision Tree (GBDT), Adaboost and PNN [35].

Combined with these studies, it is of innovative significance to apply the combination of Deep Learning, IoT and FM to medicine.

III. MODEL INTRODUCTION

The DeepFM model consists of two parts: the FM component and the Deep component, which implement low-order and high-order feature interactions, respectively.

The FM component adds a second-order feature interaction based on the linear model, and the formula is [36]:

$$y_{FM} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} x_i x_j \quad (3)$$

The first two items are the linear model part, and the last item is the second-order feature interaction part. w_0 is the bias term, x_i represents the i-th component of the feature vector $\boldsymbol{x}$, w_i represents the parameter of the i-th feature, and w_{ij} represents the parameter multiplied by the i-th and j-th features.

Categorical data is divided into three types: nominal, ordered, and distanced. For nominal variables, each value cannot be calculated and compared. Therefore, it is usually necessary to perform one-hot encoding for discrete features [37], which cause data to be particularly sparse. When the values of a series of discrete data are very large, such as 1 million, after one-hot, the feature space increases by one million dimensions, and the data will too sparse. In the second-order feature combination part, the data is too sparse and it will be impossible to learn w_{ij}. In order to solve the above problems, an implicit vector $\boldsymbol{v_i}=(v_{i1}, v_{i2},.., v_{ik})$($k$ is a hyper-parameter) is introduced for each feature of the component and replaced w_{ij} with $\boldsymbol{v_i}\,\boldsymbol{v_j}^T$. At this time, the solution of w_{ij} is transformed into the solution of v_i and v_j. The formula is now converted to [38]:

$$y_{FM} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle x_i x_j \quad (4)$$

It can be transformed, the time complexity is reduced from O(kn^2) to O(kn)[39], the transformation process is as follows:

$$\begin{aligned}
&\sum_{i=1}^{n-1} \sum_{j=i}^{n} \langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle x_i x_j \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle x_i x_j - \frac{1}{2} \sum_{i=1}^{n} \langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle x_i^2 \\
&= \frac{1}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{f=1}^{k} v_{if} v_{jf} x_i x_j - \sum_{i=1}^{n} \sum_{f=1}^{k} v_{if}^2 x_i^2 \right) \\
&= \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{if} x_i \right) \left(\sum_{j=1}^{n} v_{jf} x_j \right) - \sum_{i=1}^{n} v_{if}^2 x_i^2 \right) \\
&= \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{if} x_i \right)^2 - \sum_{i=1}^{n} v_{if}^2 x_i^2 \right) \quad (5)
\end{aligned}$$

<$\boldsymbol{v_i}$, $\boldsymbol{v_j}$> represents the inner product of the vectors $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$, and v_{if} represents the f-th component of the vector $\boldsymbol{v_i}$ [40], [41]. After such transformation and decomposition, it can be solved by Stochastic Gradient Descent (SGD) [42], The formula is as follows:

$$\frac{\partial}{\partial \theta} y(x) = \begin{cases} 1, & \text{if } \theta \text{ is } w_0 \\ x_i, & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^{n} v_{jf} x_j - v_{if} x_i^2, & \text{if } \theta \text{ is } v_{if} \end{cases} \quad (6)$$

The detailed structure of the FM part is shown in Fig. 2:

Before one-hot processing, each feature belongs to a field. In the Sparse Features section, a field of discrete data after one-hot encoding contains several columns, while continuous features remain unchanged or one column corresponds to one field. The use field can convert the sparse large matrix into a dictionary and two small matrices during storage, so as to achieve the purpose of anti-sparseness and save storage space: the small dictionary adds an index mark to each eigenvalue of the discrete feature. Each continuous feature is only given one feature index identification; the first small matrix is the feature index matrix, its length is the sample length, and each column represents the index identification of the sample feature value in the dictionary, but here is mainly considered for one. The sparsity problem caused by hot, one column replaces all the columns in a field; the second small matrix is the eigenvalue matrix, which corresponds to the eigenvalue matrix, and stores the eigenvalues of the corresponding index positions. The recorded discrete eigenvalues are all 1, instead of 0, and the continuous type stores the original value. Through the above conversion, it is convenient to find the

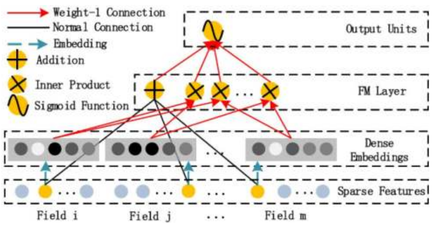


FIGURE 2. FM partial architecture diagram.

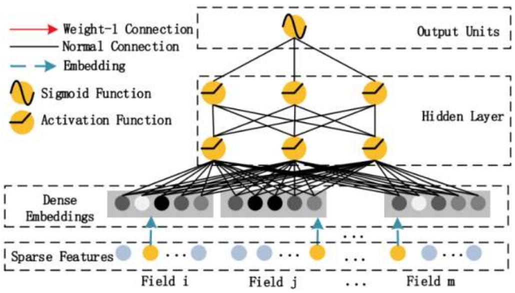


FIGURE 3. Partial architecture diagram of Deep.

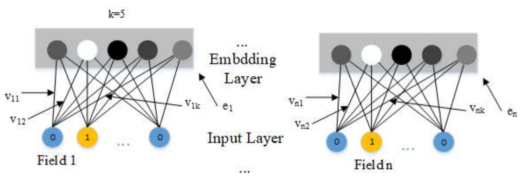


FIGURE 4. Input vector is converted into a low-dimensional dense vector.

field value of the corresponding feature to participate in the calculation.

Then introduce the Dense Embedding layer to convert the discrete vector in the Sparse Features part into a low-dimensional dense vector. This is the same as the Deep part, which will be explained below.

The FM layer part corresponds to (2), and the plus circle corresponds to the first-order feature calculation. In the figure, it is connected with Sparse Features part by a black line, and the weight is w_i; the cross circle corresponds to the interactive calculation between each feature, and the red line represents its connection with Dense Embedding that will use the aforementioned dictionary, feature index matrix, and feature value matrix to calculate interactive features.

The detailed structure of the Deep part is shown in Fig. 3:

The Deep part is a Neural Network. This part and the FM part will share the low-dimensional dense vector of the Dense Embedding part, which is transformed from the input vector of Sparse Features part. As shown in Fig. 4,

Each field will be transformed through such a fully connected network. Regardless of the length before each field (the length after one-hot), it will be transformed into a vector of length k [42]. Note that the v_{ij} here is not part of the FM Hidden vector, but here k is the same length as the hidden vector, which is convenient for subsequent calculations. All the transformed low dimensional vectors are merged, that is $\boldsymbol{a}^{(0)} = (\boldsymbol{e_1}, \boldsymbol{e_2}, \ldots, \boldsymbol{e_m})$, as the output of the Dense Embedding layer and input to the hidden layers.

Since the data processed is medical record-type structured data, DNN has shown great potential in using Electronic Health Records (EHR) to construct more accurate healthcare prediction models [43], so DNN is used as the Deep part of the model, the formula is as shown in the Wide&Deep section, the formula for the output layer is

$$\boldsymbol{a}^{(H+1)} = \sigma\left(\boldsymbol{W}^{(H)}\boldsymbol{a}^{(H)} + \boldsymbol{b}^{(H)}\right) \tag{7}$$

H is the number of hidden layers. The circle in the Hidden Layer part of Figure 2 represents the activation function relu, and the formula is

$$f(x) = \begin{cases} 0, & x \le 0 \\ x, & x > 0 \end{cases} \tag{8}$$

It can avoid the disappearance of the gradient caused by the calculation of the sigmoid function. But sigmoid usually appears in the last layer to get a probability value in the case of binary classification, the formula is [44]

$$f(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

Finally, the output of the FM part and the Deep part will be integrated in the Output Units part, and the result will be calculated by sigmoid. The formula is:

$$\hat{y} = sigmoid\left(y_{FM} + y_{Deep}\right) \tag{10}$$

The whole process is shown in Fig. 5:

However, in real life, the proportions of information contained in the combination of low-order vectors and high-order vectors are often different. In this study, a slight improvement was made to DeepFM to solve this problem by assigning different weighted hyper-parameters to linear output, second-order output and Deep part output respectively, and the calculation formula of the Output Units part is modified as follows:

$$\hat{y} = \mathrm{sigmoid}\left(\alpha * y_{Linear} + \beta * y_{Cross} + \gamma * y_{Deep}\right) \tag{11}$$

α, β and γ respectively represent different hyper-parameter values, y_{Linear}, y_{Cross}, and y_{Deep} respectively represent the linear output of the FM part, the second-order combination output, and the output of the Deep part. Subsequent experiments will prove that such improvements can improve model performance in a small range, especially the AUC value.

The IoT solution we propose is as follows: wearable sensors such as bracelets will regularly collect user weight, blood pressure, cholesterol and other easily changing data, and transmit it to smart-phones through a lightweight protocol Constrained Application Protocol (CoAP) [45]. There is a copy of the improved DeepFM model in smart-phone, where edge calculation will be performed to predict the result. If the incidence of hepatitis is too high, an alert will be issued to the user and marked as abnormal. The smart-phone will send

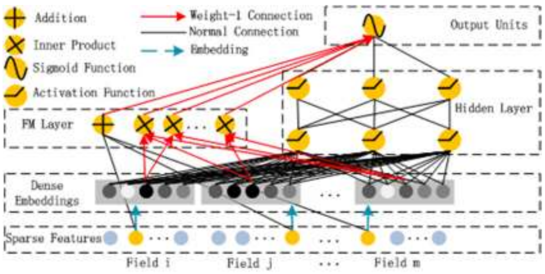


FIGURE 5. DeepFM architecture diagram.

the test results or abnormal user data to the cloud system, that is, the centralized cloud, which is stored in the cloud database for easy viewing by relevant doctors. This will reduce latency and improve bandwidth, and make full use of edge computing power [46].

By doing above approaches, a complete healthcare system has been completed.

IV. EXPERIMENT AND RESULT

A. DATA USED

Disease prediction can often be judged from the patient's medical history data mining, and many valuable discoveries are often derived from the modeling and analysis of structured data. This study uses the data from the preliminaries of the Smart Calculation 2020 Artificial Intelligence Challenge, which is a structured data set similar to EHR. It was used by JPAC Center for Health Diagnosis and Control to assess national adults in 2008-2009 and 2014-2015. It collected from two surveys of people. Through visits and research by professional medical personnel, the data set covers a wide range of population information and their health status information, which comes from direct interviews, physical examinations and blood sample examinations. In order not to affect the study, all the samples with null values for hepatitis were removed. This part only accounts for 0.25% of the total samples. In order to be consistent with the scene during the game, the first 6000 pieces of data were used as the training set, and the last 2763 pieces of data were used as the test set. This study hopes to determine whether a patient has hepatitis through structured data prediction and analysis.

The features are divided into discrete and continuous types, as shown in Table. 1. After that, the discrete data is labeled to meet the input requirements of the model; the continuous data needs to be normalized to speed up the learning speed of the model, especially when the gradient descent method is selected. Later, different dictionaries and feature index matrices will be generated for the two, and finally merged.

Due to the characteristics of DeepFM itself, it does not need to do feature processing, so the null value only needs to be filled with -1 instead of additional data cleaning work. Follow-up experimental results show that for this data set, feature engineering

TABLE 1. Features of the data set (left: discrete type, right continuous type).

Discrete features	Continuous features
ALF	Age
Family Hypertension	Weight
Gender	Height
PVD	Body Mass Index
Physical Activity	Waist
Hypertension	Maximum Blood Pressure
Region	Minimum Blood Pressure
Obesity	Good Cholesterol
Unmarried	Bad Cholesterol
Education	Total Cholesterol
Diabetes	
Chronic Fatigue	
Dyslipidemia	
Family Diabetes	
Income	

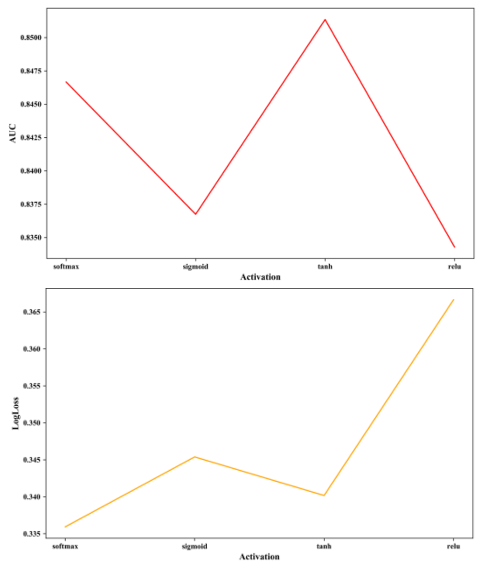


FIGURE 6. Learning curve of softmax, sigmoid, tanh, and relu activations.

It will not improve the performance of the model, but will decrease it.

B. DESCRIPTION OF EVALUATION INDICATORS

This experiment is a binary classification prediction experiment, so AUC and Logistic Loss are used for evaluation. These two evaluation indicators are the same as most evaluation indicators. The real value or predicted probability generated by the learner is used as input. For binary classification, the probability of each sample in the test set being a positive example is used as input.

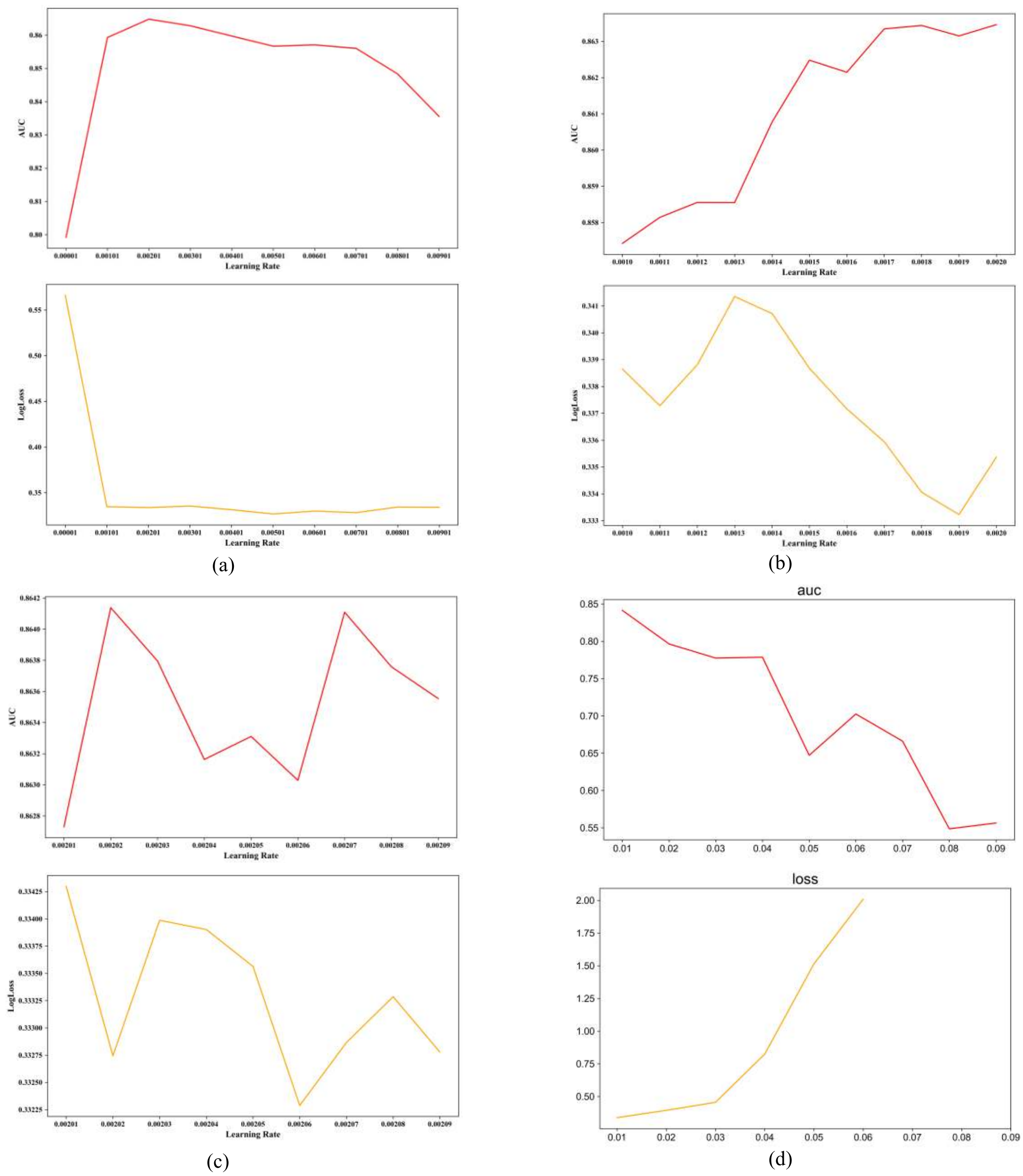


FIGURE 7. Learning rate curve. (a) range is [0.0001, 0.00901] (with 0.001 as the step size). (b) range is [0.001, 0.002] (with 0.0001 as the step size). (c) range is [0.00201, 0.00209] (with 0.00001 as the step size). (d) range is [0.01, 0.09] (with 0.01 Is the step size).

LogLoss is Logistic Loss, the formula is:

$$L = -\frac{1}{n}\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{m} y_{ij}\log p_{ij} \tag{12}$$

n represents the number of samples, m is the number of categories, y_{ij} represents 1 when the i-th sample belongs to the category j, otherwise it is 0, and p_{ij} represents the probability that the i-th sample is classified as the j-th category [47]. When it is binary classification, the formula is simplified to:

$$L = -\frac{1}{n}\sum\nolimits_{i=1}^{n}\left(y_i \log p_i + (1-y_i)\log(1-p_i)\right) \tag{13}$$

y_i is the true category of the i-th input sample x_i, and p_i is the probability of predicting that x_i is a positive sample [48]. The smaller the Logistic Loss, the better, and when it is 0, it is a perfect classifier.

The accuracy evaluation index is not suitable for samples with skewness, and the selection of the classification threshold is also a problem that needs to be considered. The AUC derived from the confusion matrix solves this problem very well. AUC is actually The top is the area under the Receiver Operating Characteristic (ROC) curve. ROC is a coordinate diagram analysis tool. The optimal threshold can be set in the model. It can remain stable when the number of positive and negative samples changes. Its graph is based on True Positive Rate (TP/TP+FN), representing the probability of dividing an actual positive sample into a positive sample) is the vertical axis, and False Positive Rate (FP/FP+TN), representing the probability of dividing an actual negative sample into a positive sample) is one of the horizontal axis curve [49]. The value of AUC is between 0-1, the closer to 1, the better [50], if it is above 0.85, it is a better classifier.

There are very few samples of hepatitis in the data set used in the experiment, which belong to the skew category. For data sets with unbalanced phenomena, AUC is mainly used as the evaluation index, and as long as the value of Logistic Loss does not exceed 0.693, it is considered a qualified model.

C. EXPERIMENTAL RESULTS

This research uses the deepctr package to implement the DeepFM model. This package is an easy-to-use and scalable deep learning click-through rate prediction algorithm package developed by Weichen Shen, a computer master from Zhejiang University and a current Alibaba's algorithm engineer. It can quickly build a CTR prediction algorithm from existing components. model. After testing, the model will have some randomness. This experiment uses the method of averaging multiple times to obtain a more accurate value, so that the model maintains a few thousandths of stability. Therefore, the results of multiple runs have a value of one thousand. The up and down fluctuation of the number is normal.

This research will first adjust the hyper-parameters of DeepFM with the learning curve, and add the weight hyper-parameter improvement scheme, and achieve the best state through analysis and adjustment, and then compare the AUC with Logistic Regression, DNN, FNN, and PNN algorithms.

1) HYPER-PARAMETER LEARNING

Initial setting: In order to reduce over-fitting, the l_2 regularization term is added to the first-order feature part and the second-order feature combination part, the penalty parameter is temporarily set to 0.00001, and a Dropout layer is added after the hidden layers, and the probability setting is discarded. It is 0.1; the learning rate is set to 0.01; the number of neurons in the Deep part is 128 and 128, respectively; the number of iterations is 10, and Mini-batch [51] is set to 128; Adam optimizer [52] is used for optimization.

Hidden layers activation function: select softmax, sigmoid, tanh, and relu activation for comparison. Both sigmoid and relu activation have been introduced above. The formula of tanh is:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{14}$$

The characteristic of tanh is that the output is centered at 0, which is faster than sigmoid; softmax is similar to sigmoid, and it also outputs a probability value between 0-1. As shown in the learning curve in Fig. 6, the AUC value of tanh is about 0.851, softmax is the second highest, and the values of softmax and tanh in the Logistic Loss graph are the lowest. Therefore, it is more appropriate to choose tanh as the activation function of the Deep hidden layers.

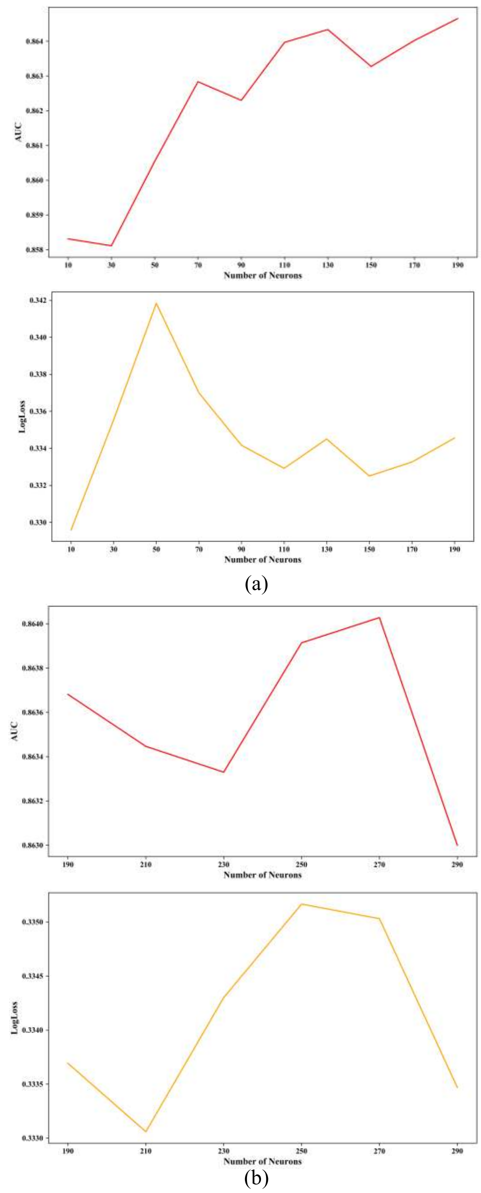
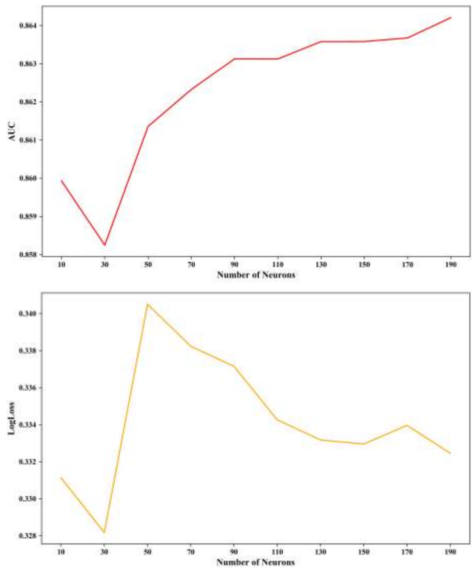


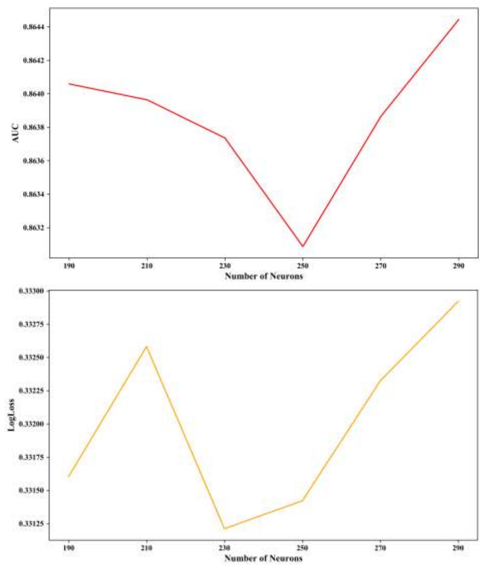
FIGURE 8. Learning curve of the number of neurons in the first hidden layers. (a) range is [10, 190] (with 20 as the step size). (b) range is [190, 290] (with 20 as the step size).

Learning rate: If the learning rate is too small, it may not be able to converge to the local optimal value (the neural network is non-convex). If the learning rate is too small, it will oscillate around the optimal value, which will never reach the optimal value. Choosing a suitable learning rate can be more effective. Accurately approximate the optimal value.

We are in [0.0001, 0.00901] (with 0.001 as the step size), [0.001, 0.002] (with 0.0001 as the step size), [0.00201, 0.00209] (with 0.00001 as the step size) and [0.01, 0.09] (with 0.01 as the step length) to test the learning rate in the interval, as shown in Fig. 7. When the learning rate exceeds 0.01, the AUC value is too low, and the Logistic Loss increases from 0.5 to about 2, which is not desirable; the Logistic Loss before 0.001 drops from 0.55 to 035. These indicate that a learning rate between 0.001-0.01 is a reasonable range. The



(a)



(b)

FIGURE 9. **Learning curve of the number of neurons in the second hidden layers. (a) range is [10, 190] (with 20 as the step size). (b) range is [190, 290] (with 20 as the step size).**

AUC and Logistic Loss curves at 0.001-0.002 have a turning point, so 0.002 is selected for follow-up learning.

Hidden layers neuron shape: There are 3 layers in the Deep part, so the number of neurons in the first two layers needs to be adjusted. The more classic way is to set both to 128. Here we still adjust. The curve of the first layer is shown in Fig. 8. It can be seen that around 128 is a more appropriate number of neurons.

The second layer is shown in Fig. 9. According to these four graphs, it can be seen that starting from 30, the AUC curve grows slightly faster, and Logistic Loss also has a significant decline. AUC continues to grow from 190 to 290, and the overall Logistic Loss is also positively correlated. On the whole, the areas near 130 and 190 are better. You can choose 190 as the number of neurons in the second layer.

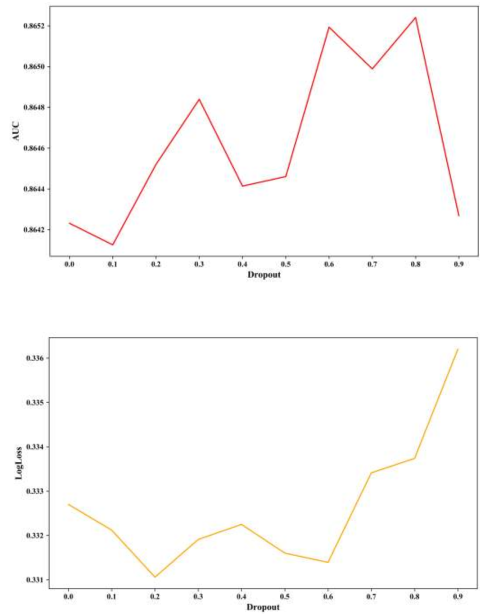


FIGURE 10. **Learning curve of dropout probability value (range is [0.0, 0.9], with 0.1 as the step size).**

Dropout: As shown in Fig. 10, choose 0.8 that makes the highest AUC value as the Dropout probability value. At this time, the AUC value of the model reaches above 0.85 for the first time.

Weight hyper-parameter improvement: adjust the weight parameters of FM linear output, second-order combined output, and Deep output. As shown in Fig. 11, it can be seen that adjusting these three parameters can indeed improve the model performance, reaching above 0.866. When the figure a is 0.8 or 1.4, the AUC reaches 0.86625; when the figure b is 0.8, the AUC reaches 0.86621; when the figure c is 1.2, the AUC reaches 0.86656. The Logistic Loss is still around 0.33. The above proves that adding this part of the improvement can indeed improve the performance of the model in a smaller range.

2) Model Comparison

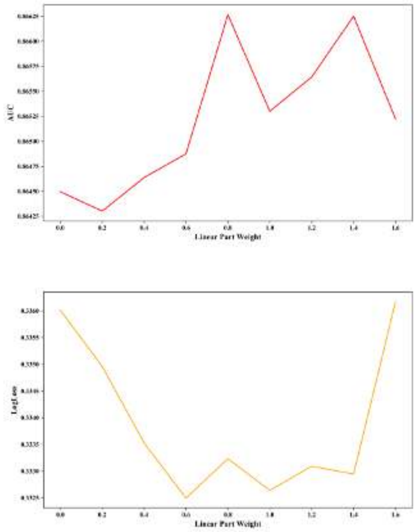
DNN and Logistic Regression are different from DeepFM and require feature engineering processing. We use the mode to fill the empty value for its discrete features, and the average value for the continuous type. The two features of Body Mass Index [53] and Total Cholesterol use formulas to fill in missing values.

$$\text{Body Mass Index} = \text{Weight (kg)} / \text{Height}(\text{m}^2) \tag{15}$$

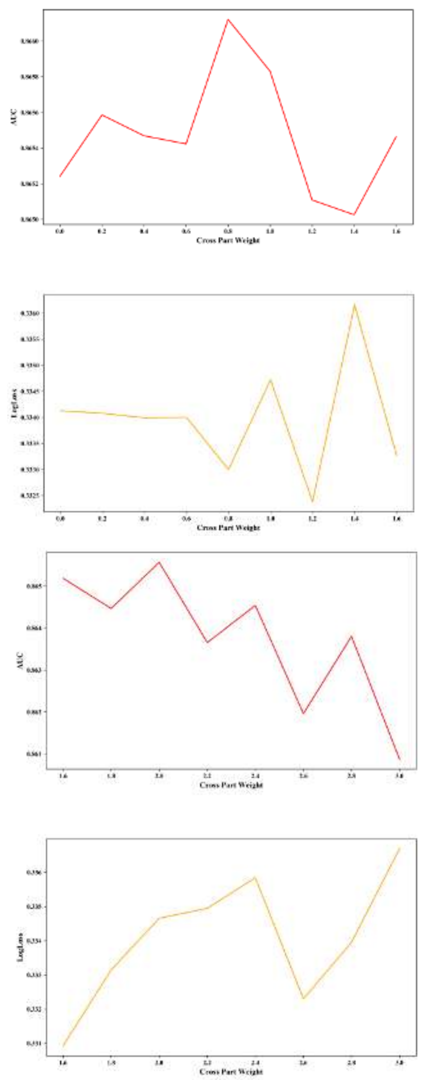
$$\text{TotalCholesterol} = (\text{Good Cholesterol+Bad Cholesterol}) \tag{16}$$

For better and faster convergence of the model, the features are also normalized.

DNN was introduced earlier. As shown in Fig. 12, increasing the number of layers to more than 4 will not improve, but will decrease, increasing model complexity. Therefore, a 3-layer DNN is used here to compare with other models, and the hyper-parameter values adjusted by the learning curve

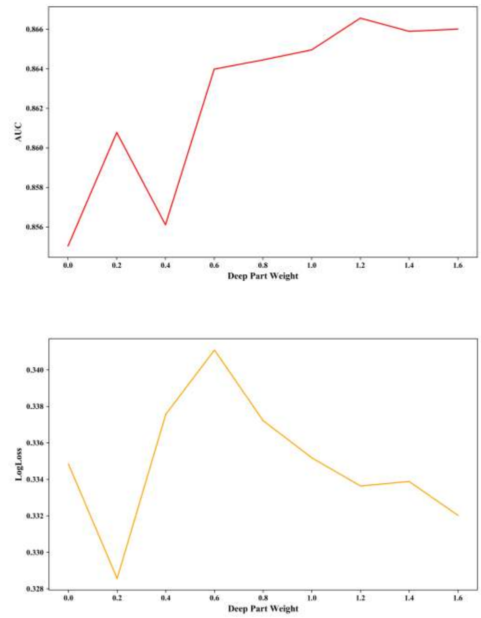


(a)



(b)

FIGURE 11. **Weight hyper-parameter curve. (a) Linear output weight parameter curve of the FM part. (b) Second-order combined output weight parameter curve. (c) The output weight parameter curve of the Deep part.**



(c)

FIGURE 11. ***(Continued.)*** **Weight hyper-parameter curve. (a) Linear output weight parameter curve of the FM part. (b) Second-order combined output weight parameter curve. (c) The output weight parameter curve of the Deep part.**

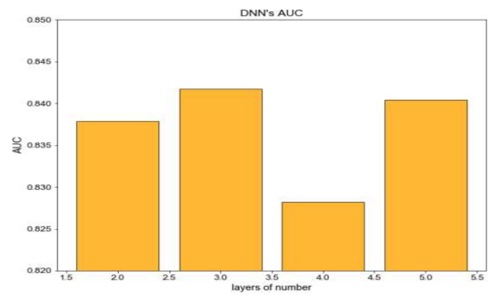


FIGURE 12. **Comparison of DNN layers.**

of the 3 layers are listed: the learning rate is 0.00773; the activation functions of the first two hidden layers are set to tanh, and the two layers of neural The number of elements are 128 and 139 respectively; because it is binary classification, the activation function of the output layer is sigmoid; the number of iterations is 21; in order to prevent over fitting, both the hidden layers and the output layer are l2 regularized. Add a Dropout layer with a probability of 0.3 between the two hidden layers.

Logistic Regression: Logistic Regression is a probabilistic nonlinear regression model, the formula is sigmoid function, the difference is that its independent variable is the result of linear regression [54], that is, the FM model removes the second-order interactive part, leaving only the linear part. The model can be expressed as:

$$P\left(y=1|x\right)=\frac{1}{1+e^{-g(x)}} \tag{17}$$

TABLE 2. Comparison table of optimal AUC for 3-layer DNN, Logistic, DeepFM, FNN, and PNN.

	AUC
3-layer DNN	0.8417451261063285
Logistic Regression	0.8596661357633918
DeepFM	0.8665605532146529
FNN	0.865526098264148
PNN	0.8497860930223181

and its loss function is the Logistic Loss two-category form (Equation 13). Calculated by liblinear, the adjustment parameter value is: the number of iterations is 8, and l_2 regularization is added. Since the python machine learning package Scikit-learn is used for implementation, the penalty coefficient is added to the loss function instead of the regularization term. The penalty coefficient is set to 16, which is to reduce the proportion of the regularization term accordingly.

The performance comparison of 3-layer DNN, Logistic, DeepFM, FNN, and PNN is shown in Table. 2. In fact, the average AUC of DeepFM is above 0.865, which also exceeds the average AUC of other models.

After joining the IoT of edge computing, users can be notified of abnormal situations in time.

V. CONCLUSION

In this article, we use the FM-based neural network model that was originally applied to the CTR problem, DeepFM, to predict the presence or absence of hepatitis in this data set sample based on the structured data for disease prediction in the preliminaries of the Smart Calculation 2020 Artificial Intelligence Challenge. On the issue of probability, the introduction of the Internet of Things mechanism facilitates remote diagnosis and forms a healthcare system. DeepFM consists of the FM part and the Deep part, and has the following advantages: first, it can learn low-level and high-level feature interactions at the same time. Second, the FM part and the Deep part share the same input and do not require pre-training and feature engineering. In practical applications, the proportions of low-level feature interaction and high-level feature interaction information are different. Therefore, we improve the original model and give different weights to the linear output, the second-order combined output and the output of Deep part respectively. Experiments show that the improved DeepFM is slightly higher in AUC than the original DeepFM, with an average value of 0.865 or more, and a maximum value of 0.8665, and both are higher than the models listed. The final system can also meet user requirements. The combination of DeepFM and the Internet of Things is instructive for disease prediction.

In the future, we will introduce the attention mechanism to be applied to the field of disease prediction. The attention mechanism, that is, Attentional Factorization Machine (AFM) will learn the importance of each feature interaction from the data; the high-order cross-factoring function can model the weights of different interaction orders [55], so that the model has good interpretability, which is also one of our research directions in the future.

REFERENCES

[1] Q. Zhang, J. Lu, and Y. Jin, "Artificial intelligence in recommender systems," *Complex Intell. Syst.*, vol. 7, no. 1, pp. 439–457, Feb. 2021.

[2] W. T. Le, F. Maleki, F. P. Romero, R. Forghani, and S. Kadoury, "Overview of machine learning: Part 2: Deep learning for medical image analysis," *Neuroimag. Clinics North Amer.*, vol. 30, no. 4, pp. 417–431, 2020.

[3] R. J. Erb, "Introduction to backpropagation neural network computation," *Pharmaceutical Res.*, vol. 10, no. 2, pp. 165–170, 1993.

[4] M. Ben-Ari and F. Mondada, "Neural networks," in *Elements of Robotics*. Cham, Switzerland: Springer, 2018, pp. 203–220.

[5] M. F. Alhamid *et al.*, "Towards context-sensitive collaborative media recommender system," *Multimedia Tools Appl.*, vol. 74, no. 24, pp. 11399–11428, Dec. 2015.

[6] X. Lin, W. Zhang, M. Zhang, W. Zhu, J. Pei, P. Zhao, and J. Huang, "Online compact convexified factorization machine," in *Proc. World Wide Web Conf. World Wide Web WWW*, 2018, pp. 1633–1642.

[7] Y. Zhang, X. Zhu, and Q. Shen, "A recommendation model based on collaborative filtering and factorization machines for social networks," in *Proc. 5th IEEE Int. Conf. Broadband Netw. Multimedia Technol.*, Nov. 2013, pp. 110–114.

[8] S. Rendle, "Factorization machines with libFM," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 1–22, May 2012.

[9] Z. Jiang, S. Gao, and M. Li, "An improved advertising CTR prediction approach based on the fuzzy deep neural network," *PLoS ONE*, vol. 13, no. 5, May 2018, Art. no. e0190831.

[10] H. Julkunen, A. Cichonska, P. Gautam, S. Szedmak, J. Douat, T. Pahikkala, T. Aittokallio, and J. Rousu, "Leveraging multi-way interactions for systematic prediction of pre-clinical drug combination effects," *Nature Commun.*, vol. 11, no. 1, p. 6136, Dec. 2020.

[11] L. Zhang, W. Shen, J. Huang, S. Li, and G. Pan, "Field-aware neural factorization machine for click-through rate prediction," *IEEE Access*, vol. 7, pp. 75032–75040, 2019.

[12] J. Pan, J. Xu, A. L. Ruiz, W. Zhao, S. Pan, Y. Sun, and Q. Lu, "Field-weighted factorization machines for click-through rate prediction in display advertising," in *Proc. World Wide Web Conf. World Wide Web - WWW*, 2018, pp. 1349–1357.

[13] J. Lin *et al.*, "Deep-cross-attention recommendation model for knowledge sharing micro learning service," in *Artificial Intelligence in Education* (Lecture Notes in Computer Science). 2020, pp. 168–173.

[14] P. Wen, W. Yuan, Q. Qin, S. Sang, and Z. Zhang, "Neural attention model for recommendation based on factorization machines," *Appl. Intell.*, Oct. 2020.

[15] R. Hou, Y. Kong, B. Cai, and H. Liu, "Unstructured big data analysis algorithm and simulation of Internet of Things based on machine learning," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 5399–5407, May 2020.

[16] M. S. Hossain and G. Muhammad, "Deep learning based pathology detection for smart connected healthcares," *IEEE Netw.*, vol. 34, no. 6, pp. 120–125, Nov./Dec. 2020.

[17] M. A. Rahman and M. S. Hossain, "An Internet of medical things-enabled edge computing framework for tackling COVID-19," *IEEE Internet Things J.*, early access, Jan. 12, 2021, doi: 10.1109/JIOT.2021.3051080.

[18] L. Vesnic-Alujevic, M. Breitegger, and Â. G. Pereira, "'Do-it-yourself' healthcare? Quality of health and healthcare through wearable sensors," *Sci. Eng. Ethics*, vol. 24, no. 3, pp. 887–904, Jun. 2018.

[19] Y. Zhang, X. Ma, J. Zhang, M. S. Hossain, G. Muhammad, and S. U. Amin, "Edge intelligence in the cognitive Internet of Things: Improving sensitivity and interactivity," *IEEE Netw.*, vol. 33, no. 3, pp. 58–64, May 2019, doi: 10.1109/MNET.2019.1800344.
[20] M. Hosseinzadeh *et al.*, "A multiple multilayer perceptron neural network with an adaptive learning algorithm for thyroid disease diagnosis in the Internet of medical things," *J. Supercomput.*, Aug. 2020.
[21] M. S. Hossain, G. Muhammad, and A. Alamri, "Smart healthcare monitoring: A voice pathology detection paradigm for smart cities," *Multimedia Syst.*, vol. 25, no. 5, pp. 565–575, Oct. 2019.
[22] S. Sharma, R. K. Dudeja, G. S. Aujla, R. S. Bali, and N. Kumar, "DeTrAs: Deep learning-based healthcare framework for IoT-based assistance of Alzheimer patients," *Neural Comput. Appl.*, 2020.
[23] M. S. Hossain, "Cloud-supported cyber–physical localization framework for patients monitoring," *IEEE Syst. J.*, vol. 11, no. 1, pp. 118–127, Mar. 2017.
[24] J. P. Rajan, S. E. Rajan, R. J. Martis, and B. K. Panigrahi, "Fog computing employed computer aided cancer classification system using deep neural network in Internet of Things based healthcare system," *J. Med. Syst.*, vol. 44, no. 2, p. 34, Dec. 2019.
[25] M. S. Hossain and G. Muhammad, "Cloud-based collaborative media service framework for healthcare," *Int. J. Distrib. Sens. Netw.*, vol. 10, no. 3, Mar. 2014, Art. no. 858712.
[26] S. U. Amin, M. Alsulaiman, G. Muhammad, M. A. Bencherif, and M. S. Hossain, "Multilevel weighted feature fusion using convolutional neural networks for EEG motor imagery classification," *IEEE Access*, vol. 7, pp. 18940–18950, 2019, doi: 10.1109/ACCESS.2019.2895688.
[27] C. Yan, Y. Chen, Y. Wan, and P. Wang, "Modeling low- and high-order feature interactions with FM and self-attention network," *Appl. Intell.*, Nov. 2020.
[28] X. Yang *et al.*, "Deep relative attributes," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1832–1842, Sep. 2016.
[29] D. Zou, M. Sheng, H. Yu, J. Mao, S. Chen, and W. Sheng, "Factorized weight interaction neural networks for sparse feature prediction," *Neural Comput. Appl.*, vol. 32, no. 13, pp. 9567–9579, Jul. 2020.
[30] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 355–364.
[31] Y. Yu, L. Jiao, N. Zhou, L. Zhang, and H. Yin, "Enhanced factorization machine via neural pairwise ranking and attention networks," *Pattern Recognit. Lett.*, vol. 140, pp. 348–357, Dec. 2020.
[32] L. Bo, Z. Xiao, L. Kewei, and Y. Jingyi, "Analysis of the click through rate of Chinese Netizens through social software based on DeepFM algorithm," *J. Phys., Conf. Ser.*, vol. 1284, Aug. 2019, Art. no. 012025.
[33] Z. Zhou, L. Zhao, A. Lin, W. Qin, Y. Lu, J. Li, Y. Zhong, and L. He, "Exploring the potential of deep factorization machine and various gradient boosting models in modeling daily reference evapotranspiration in China," *Arabian J. Geosci.*, vol. 13, no. 24, pp. 1–20, Dec. 2020.
[34] M. Yamada, W. Lian, A. Goyal, J. Chen, K. Wimalawarne, S. A. Khan, S. Kaski, H. Mamitsuka, and Y. Chang, "Convex factorization machine for toxicogenomics prediction," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1215–1224.
[35] Y. Fan, D. Li, Y. Liu, M. Feng, Q. Chen, and R. Wang, "Toward better prediction of recurrence for Cushing's disease: A factorization-machine based neural approach," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 3, pp. 625–633, Mar. 2021.
[36] Y. Huo, "Agricultural product E-commerce recommendation system based on neural factorization machine in smart cities," *Internet Technol. Lett.*, vol. 4, no. 1, Jan. 2021, Art. no. e216.
[37] V. K. Singh, N. S. Maurya, A. Mani, and R. S. Yadav, "Machine learning method using position-specific mutation based classification outperforms one hot coding for disease severity prediction in haemophilia 'A,'" *Genomics*, vol. 112, no. 6, pp. 5122–5128, Nov. 2020.
[38] S. Qu, G. Guo, Y. Liu, Y. Yao, and W. Wei, "Fast discrete factorization machine for personalized item recommendation," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105470.
[39] M. Tang, W. Liang, Y. Yang, and J. Xie, "A factorization machine-based QoS prediction approach for mobile service selection," *IEEE Access*, vol. 7, pp. 32961–32970, 2019.
[40] C. Chen, W. Dongxing, H. Chunyan, and Y. Xiaojie, "Exploiting social media for stock market prediction with factorization machine," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. (WI) Intell. Agent Technol. (IAT)*, Aug. 2014.
[41] F. Yuan, G. Guo, J. M. Jose, L. Chen, H. Yu, and W. Zhang, "BoostFM: Boosted factorization machines for Top-N feature-based recommendation," in *Proc. 22nd Int. Conf. Intell. User Interface*, Mar. 2017, pp. 45–54.
[42] M. Zhu, C. C. Aggarwal, S. Ma, H. Zhang, and J. Huai, "Outlier detection in sparse data with factorization machines," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 817–826.
[43] M. Alhussein, G. Muhammad, M. S. Hossain, and S. U. Amin, "Cognitive IoT-cloud integration for smart healthcare: Case study for epileptic seizure detection and monitoring," *Mobile Netw. Appl.*, vol. 23, no. 6, pp. 1624–1635, Dec. 2018, doi: 10.1007/s11036-018-1113-0.
[44] H. Zhao, T. Mao, J. Duan, Y. Wang, and H. Zhu, "FMCNN: A factorization machine combined neural network for driving safety prediction in vehicular communication," *IEEE Access*, vol. 7, pp. 11698–11706, 2019.
[45] E. Adi, A. Anwar, Z. Baig, and S. Zeadally, "Machine learning and data analytics for the IoT," *Neural Comput. Appl.*, vol. 32, no. 20, pp. 16205–16233, 2020.
[46] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, and J. Qin, "A survey on application of machine learning for Internet of Things," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 8, pp. 1399–1417, 2018.
[47] X. Zhang, J. Zhong, and K. Liu, "Wasserstein autoencoders for collaborative filtering," *Neural Comput. Appl.*, 2020.
[48] H. Li, H. Duan, Y. Zheng, Q. Wang, and Y. Wang, "A CTR prediction model based on user interest via attention mechanism," *Int. J. Speech Technol.*, vol. 50, no. 4, pp. 1192–1203, Apr. 2020.
[49] S. Rosset, "Model selection via the AUC," presented at the 21st Int. Conf. Mach. Learn., Banff, AB, Canada, 2004, doi: 10.1145/1015330.1015400.
[50] H. Hong, B. Pradhan, M. I. Sameen, W. Chen, and C. Xu, "Spatial prediction of rotational landslide using geographically weighted regression, logistic regression, and support vector machine models in xing guo area (China)," *Geomatics, Natural Hazards Risk*, vol. 8, no. 2, pp. 1997–2022, Dec. 2017.
[51] G. Danner and M. Jelasity, "Fully distributed privacy preserving mini-batch gradient descent learning," in *Distributed Applications and Interoperable Systems*. Cham, Switzerland: Springer, 2015, pp. 30–44.
[52] M. N. Halgamuge, E. Daminda, and A. Nirmalathas, "Best optimizer selection for predicting bushfire occurrences using deep learning," *Natural Hazards*, vol. 103, no. 1, pp. 845–860, Aug. 2020.
[53] M. Garrouste-Orgeas, G. Troché, E. Azoulay, A. Caubel, A. de Lassence, C. Cheval, L. Montesino, M. Thuong, F. Vincent, Y. Cohen, and J. F. Timsit, "Body mass index. An additional prognostic factor in ICU patients," *Intensive Care Med*, vol. 30, no. 3, pp. 437–443, Mar. 2004.
[54] K. Zhao, J. Zhang, L. Zhang, C. Li, and H. Chen, "A distributed coordinate descent algorithm for learning factorization machine," in *Advances in Knowledge Discovery and Data Mining*. Cham, Switzerland: Springer, 2020, pp. 881–893.
[55] W. Zhang, X. Zhang, and H. Wang, "High-order factorization machine based on cross weights network for recommendation," *IEEE Access*, vol. 7, pp. 145746–145756, 2019.
[56] W. Min, B.-K. Bao, C. Xu, and M. S. Hossain, "Cross-platform multi-modal topic modeling for personalized inter-platform recommendation," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1787–1801, Oct. 2015.
[57] M. S. Hossain and G. Muhammad, "Cloud-assisted industrial Internet of Things (IIoT)—Enabled framework for health monitoring," *Comput. Netw.*, vol. 101, pp. 192–202, Jun. 2016.
[58] L. Hu, M. Qiu, J. Song, M. S. Hossain, and A. Ghoneim, "Software defined healthcare networks," *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 67–75, Dec. 2015.
[59] G. Muhammad, M. S. Hossain, and N. Kumar, "EEG-based pathology detection for home health monitoring," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 603–610, Feb. 2021.
[60] L. Yue, D. Tian, W. Chen, X. Han, and M. Yin, "Deep learning for heterogeneous medical data analysis," *World Wide Web*, vol. 23, no. 5, pp. 2715–2737, Sep. 2020.
[61] M. S. Hossain and G. Muhammad, "Emotion-aware connected healthcare big data towards 5G," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2399–2406, Aug. 2018.
[62] S. Qian, T. Zhang, C. Xu, and M. S. Hossain, "Social event classification via boosted multimodal supervised latent Dirichlet allocation," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 11, no. 2, pp. 27.1–27.22, Jan. 2015.