Hindawi
Scientific Programming
Volume 2021, Article ID 1492828, 8 pages
https://doi.org/10.1155/2021/1492828

Research Article

Learning Deep Attention Network from Incremental and Decremental Features for Evolving Features

Chuxin Wang[1] **and Haoran Mo**[2]

[1]*Hebei University of Environmental Engineering, Qin Huang Dao 066000, China*
[2]*Innopolis University, Innopolis, Russia*

Correspondence should be addressed to Chuxin Wang; wangchuxin@hebuee.edu.cn

Received 5 May 2021; Revised 30 May 2021; Accepted 8 June 2021; Published 17 June 2021

Academic Editor: Bai Yuan Ding

In many real-world machine learning problems, the features are changing along the time, with some old features vanishing and some other new features augmented, while the remaining features survived. In this paper, we propose the cross-feature attention network to handle the incremental and decremental features. This network is composed of multiple cross-feature attention encoding-decoding layers. In each layer, the data samples are firstly encoded by the combination of other samples with vanished/augmented features and weighted by the attention weights calculated by the survived features. Then, the samples are encoded by the combination of samples with the survived features weighted by the attention weights calculated from the encoded vanished/augmented feature data. The encoded vanished/augmented/survived features are then decoded and fed to the next cross-feature attention layer. In this way, the incremental and decremental features are bridged by paying attention to each other, and the gap between data samples with a different set of features are filled by the attention mechanism. The outputs of the cross-feature attention network are further concatenated and fed to the class-specific attention and global attention network for the purpose of classification. We evaluate the proposed network with benchmark data sets of computer vision, IoT, and bio-informatics, with incremental and decremental features. Encouraging experimental results show the effectiveness of our algorithm.

1. Introduction

1.1. Background. In the machine learning problems, a basic assumption is the data samples have consistent and stable features. These features are usually generated by a set of sensors and used by the machine learning models as inputs. However, in many real-world applications, this assumption does not hold, and the features are changing with some old features vanishing and some new features added. For example, in the application of environmental monitoring, different sensors are deployed, including gravimetric, optical, and electrochemical sensors [1–4]. These sensors have different life cycle lengths and different working conditions. Thus, some sensors expired sooner than the others; thus, the corresponding features vanished sooner. Meanwhile, some other sensors can be used for a long time to continue to generate features. Their features will be surviving along the data collection process. Moreover, with the development of sensors, some new sensors are produced and deployed and begin to generate newly augmented features. As a result, the working sensors are evolving over time and the features are changing accordingly. Some old features are vanishing and some new features are augmented, while the remaining features survive. This scenario makes the feature not stable and challenges the stable feature assumption of most popular machine learning settings [5–8]. This problem is called the incremental and decremental feature (IDF) problem. Given the importance of the IDF problem, surprisingly, only very few works have been done to solve it directly [6], and the performance is not satisfying.

Meanwhile, the deep attention network has been a popular method for the machine learning area. Attention mechanism represents a data instance not only by itself but also by paying attention to the other instances weighted by

the attention weights. The attention weights are usually calculated according to the instance features and then normalized by the softmax function [9–12]. There are two types of attention network: self-attention [13–16] and cross-attention networks [17–20]. The self-attention mechanism calculates the attention weight of each instance from itself, while the cross-attention mechanism usually calculates attention weights according to the similarity between itself and the other instances. However, most existing attention network only pays attention to instances and assuming the features are stable. Thus, the attention mechanism of existing methods cannot be applied to the IDF problem.

In this paper, we propose a novel solution for the IDF problem with a cross-feature attention network. Our solution pays attention to the vanished, survived, and augmented features to bridge the gaps among the features of evolving sensors. This is the first work of attention mechanism across features, and it fits the nature of the IDF problem.

1.2. Related Works. In this section, we review the related works of IDF; even there are only very few such existing works.

(i) Hou and Zhou [6] developed an algorithm to handle the incremental and decremental features and the streaming data instances. This algorithm has two stages. The first stage is to compress the vanished features by learning a classifier in the vanished feature space so that the important information of the vanished features is embedded in the trained classifier. The second stage is an expanding stage. It will not only include the augmented features in this stage but also try to balance the vanished features, survived features, and augmented features. The balancing is conducted by imposing the classification responses with/without the augmented features. Moreover, the learning strategy is one-pass learning, which takes only one training sample to update the model in each iteration.

(ii) Ma et al. [7] designed a transfer learning method for the domains, where only a part of the features are shared, while the other features are different. This problem setting is similar to the IDF, given the partially shared features space. To be specific, the target domain not only has the source domain's feature but also has some newly augmented features. To solve this problem, this method also imposes the target domain's classification responses of data with/without augmented features to be consistent with each other. Moreover, the features shared across the source and target domains are also jointly regularized to be consistently sparse, i.e., the importance of the same feature should be consistent across two domains.

(iii) Wu et al. [21] proposed a feature selection algorithm to handle the streaming features. In this scenario, the features are not known from the very beginning of the learning process, but come in a one-by-one way, while the number of training samples remains the same. The algorithm is designed to select the most important features from the streaming feature set. The selected features should be not only relevant but also nonredundant. The feature selection is performed in an iterative algorithm. When the algorithm receives a new feature, the algorithm first determines if it is relevant to the class. If not relevant and is also redundant, it will be dropped. Otherwise, this feature is selected. The problem of streaming features is a special case of the IDF, where it only handles the streaming incremental features but ignores the decremental features.

Among these existing methods, the IDF problem is solved by imposing consistency of classification responses with/without augmented features, in both works of [6, 7]. The intersection of vanished/augmented/survived features is not explored directly. Thus, the cross-feature information is not utilized effectively to boost the learning performance.

1.3. Our Contribution. To fill this gap, in this paper, we propose the first attention network to pay attention from one feature set to another one. The motivation to do so is that we believe even the feature changes in the sample batches collected from a different time, and they have an inner relationship and they are complimentary for the purpose of classification of the samples. The sensor evolving changes the observed features, but actually, the features should be complete and consistent in an ideal situation where all the sensors do not expire and are all deployed at the very beginning. Thus, we would like to recover the vanished features for the new batch of data, and also recover the augmented features for the old batch of data. For this purpose, we encode each sample by paying attention to the vanished/augmented features. To explore the feature relationship, we calculate the attention weight by the survived features. In this way, we have a vanished/augmented feature-attention code vector for each sample, and even it has no vanished/augmented features, by bridging itself to the samples with vanished/augmented features with help of survived-feature attention. With the vanished/augmented feature-attention code vectors, we pay attention back to the survived features by encoding each sample as the combination of other samples' survived features. The attention weights are again calculated by the codes of the last cross-attention layers. Decoders are also applied to recover the original features from the code vectors, and the recovered features are inputs of the next cross-attention layers. In this way, we design a deep cross-feature attention network to represent the samples with IDF. The encoded vectors of the network are further represented by a set of class-specific attention networks and a global attention network for the purpose of classification.

Our contribution is threefold:

(1) We design a novel deep neural network with new cross-attention layers for the purpose of learning from evolving

(2) We propose a novel learning algorithm to optimize the parameters of the network in a supervised way

(3) We evaluate the proposed algorithm experimentally regarding parameter sensitivity, running time, and comparison to other algorithms

1.4. Paper Organization. This paper is organized as follows. In Section 2, we describe the new network of cross-feature attention. In Section 3, we evaluate the proposed method experimentally. In Section 4, the conclusion is given.

2. Cross-Feature Attention Network

2.1. Problem Setting. In this section, we discuss a learning problem with changing features. Suppose we have training set of n data samples, and these samples belong to two batches. One batch has no data samples generated from a set of historical sensors, and the other batch has $n_c = n - n_o$ data samples generated from the current set of sensors. Compare to the current sensor set, some old sensors have vanished, some new senors have been added to the current sensor set, and the remaining sensors remain the same. The old data batch is denoted as $X^o = \{(x_i, y_i, \psi_i)\}_{i=1}^{n_o}$, where $x_i \in R^{d1}$ is the ith sample's feature vector of the d_1 vanished sensors, $y_i \in R^{d2}$ is its feature vector of the d_2 survived sensors, and $\psi_i \in \{1, \ldots, C\}$ is its class label. The current data batch is denoted as $Xc = \{y_i, z_i, \psi_i\}_{i=n_o+1}^{n}$, where $y_i \in R^{d2}$ is the ith sample's feature vector of the d_2 survived sensors, while $z_i \in R^{d3}$ is its feature vector of the d_3 newly added sensors, and ψ_i is its class label. The overall training data set is $X = Xo \cup Xc$, and the learning problem is to learn a model to predict the class label of a test data sample with features of the current sensors.

2.2. Network Architecture. To represent each data sample of both current and old batches, we propose a deep cross-feature attention representation network and a discriminative network to separate samples of different classes.

2.2.1. Cross-Feature Attention Layers. Given the ith data sample, to represent it, we propose to pay attention from itself to three feature spaces, which are the vanished sensor space, survived sensor space, and newly added sensor space.

(1) Attention to Vanished Sensor Space. We firstly pay attention from the ith sample to the old batch Xo, even the ith sample is from the current batch. To this end, we calculate its similarity to the jth sample of Xo in the feature space of the survived sensors shared by both batches. The similarity between the ith and jth sample is calculated as

$$s\left(y_i, y_j; A\right) = y_i^\top A y_j, \quad j \in Xo, \tag{1}$$

where $A \in R^{d2 \times d2}$ is the parameter matrix of the similarity function. The attention score from the ith sample to the jth sample regarding the survived sensors is obtained by applying a softmax function to the similarity scores:

$$\alpha_{ij} = \frac{\exp\left(s\left(y_i, y_j; A\right)\right)}{\sum_{j\prime \in Xo} \exp\left(s\left(y_i, y_{j\prime}; A\right)\right)}. \tag{2}$$

With the attention scores calculated from survived sensor features, we represent the ith sample by combining the transformed features of the vanished sensor features weighted by these attention scores:

$$f_i = \sum_{j:\ j \in Xo} \alpha_{ij} \Theta^\top x_j, \tag{3}$$

where Θ is the transforming matrix. Please note, in this attention-based representation of the ith sample, the attention scores are calculated in the survived sensor space, while the attention base vectors are in the vanished sensor features space.

(2) Attention to Newly Added Sensor Space. We also pay attention to the current batch of training samples in the space of newly added sensors. To this end, we calculate the attention weights in the space of the survived sensors and use them to weigh the samples of the current batch in the new sensor space. We firstly calculate the similarity between the ith sample ($i \in X$) and the jth sample of the current batch:

$$s\left(y_i, y_j; B\right) = y_i^\top B y_j, \quad j \in Xc, \tag{4}$$

where B is the similarity function parameter matrix. Accordingly, we apply a softmax function to the similarities to calculate the attention weights:

$$\beta_{ij} = \frac{\exp\left(s\left(yi, yj; B\right)\right)}{\sum_{j\prime \in Xc} \exp\left(s\left(yi, yj\prime; B\right)\right)}. \tag{5}$$

The new representations of the ith sample by the attention to the current training batch in the space of newly added sensor space is the combination of the transformed samples with the above weights:

$$h_i = \sum_{j:\ j \in Xc} \beta_{ij} \Phi^\top z_j, \tag{6}$$

where Φ is the transforming parameter matrix. The cross-feature mapping is performed from newly added features with weights of the survived features.

(3) Attention to Survived Sensor Space. Paying attention to the samples of the entire data set is weighted by the representations of the above two layers. Given the ith sample, we firstly concatenate the two vectors of the last two attention layers, f_i and h_i, to a longer vector, $t_i = \begin{bmatrix} f_i \\ h_i \end{bmatrix} \in R^{d1+d3}$. With this vector, we calculate the similarity between two samples, the ith and jth samples:

$$s\left(ti, tj; E\right) = t_i^\top E t_j, \quad j \in X, \tag{7}$$

where E is the similarity function parameter matrix. The attention weights are calculated by softmax:

$$\gamma_{ij} = \frac{\exp\left(s\left(t_i, t_j; E\right)\right)}{\sum_{j\prime \in X} \exp\left(s\left(t_i, t_{j\prime}; E\right)\right)}. \tag{8}$$

The attention layer output vector of the ith sample is the combination of the features of survived sensors weighted by attention weights in (8):

$$g_i = \sum_{j:\ j \in X} \gamma_{ij} \Psi^\top y_j, \tag{9}$$

where Ψ is the transforming matrix.

(4) Decoding Layer. With the above three layers of cross-attentions, we have three representation vectors f_i, g_i, and h_i. We can further decode the sensor features from these vectors for the next layers' inputs in a deep network architecture. The decoding layers are dense layers with activation layers:

$$\begin{aligned} x_i^{\text{new}} &= \varphi\left(W^\top f_i\right), \quad \forall i \in Xo, \\ y_i^{\text{new}} &= \varphi\left(V^\top g_i\right), \quad \forall i \in X, \\ z_i^{\text{new}} &= \varphi\left(R^\top h_i\right), \quad \forall i \in Xc, \end{aligned} \tag{10}$$

where W, V, and R are the dense layer parameter matrices and $\varphi(\cdot)$ is the activation function.

Given the above base layers, we build a multiple layer cross-feature attention network by feeding the outputs of the decoding layer to the next layers of old and newly added sensor attention layers. In the lth layer, the output of the $l-1$th layer is x_i^{l-1}, y_i^{l-1}, and z_i^{l-1} for the lth sample, and itwill be used to estimate f_i^l and h_i^l of this layer according to (3) and (6):

$$\begin{aligned} f_i^l &= \sum_{j:\ j \in Xo} \alpha_{ij}^l \Theta_l^\top x_j^{l-1}, \\ h_i^l &= \sum_{j:\ j \in Xc} \beta_{ij}^l \Phi_l^\top z_j^{l-1}. \end{aligned} \tag{11}$$

Then, f_l iand f_i^l will be used to recalculate the weights of (8), γ_{ij}^l, and finally estimate g_i^l as

$$g_i^l = \sum_{j:\ j \in X} \gamma_{ij}^l \Psi_l^\top y_j^{l-1}. \tag{12}$$

The decoding layer is applied to generate the outputs of this layer:

$$\begin{aligned} x_i^l &= \varphi\left(W_l^\top f_i^l\right), \quad \forall i \in Xo, \\ y_i^l &= \varphi\left(V_l^\top g_i^l\right), \quad \forall i \in X, \\ z_i^l &= \varphi\left(R_l^\top h_i^l\right), \quad \forall i \in Xc. \end{aligned} \tag{13}$$

The cross-feature attention layer is shown in Figure 1. We can see that, in this layer, the input data has three sets of features, and the attention is paid from one feature to another. To be more specific, the new presentation of one feature is the combination of samples in this feature space, but the weights of attention are estimated from another feature space.

Suppose we have L layers of cross-feature attention and the outputs of the last attention layers are f_i^L, g_i^L, and h_i^L. In our implementation, we set the layer number L to 12. They

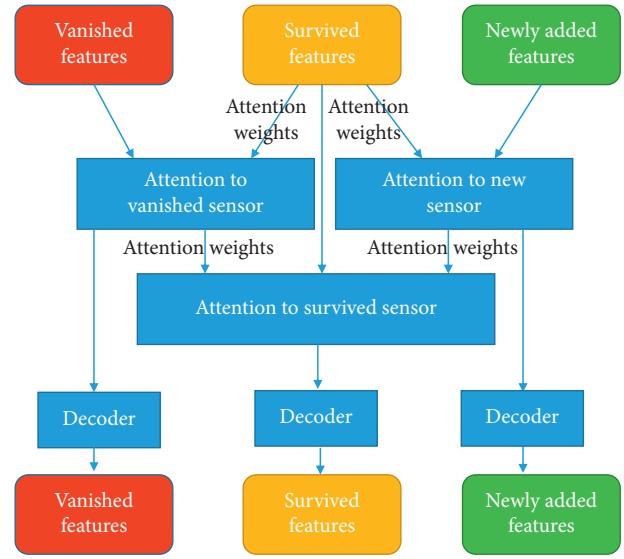

Figure 1: Cross-feature attention layer.

are further concatenated as a long vector to represent the ith sample as follows:

$$u_i = \begin{bmatrix} f_i^L \\ g_i^L \\ h_i^L \end{bmatrix}. \tag{14}$$

This vector will be the input of the next class-specific attention network for the purpose of classification.

2.2.2. Class-Specific Attention Layers. Given the ith samples cross-feature attention representation, u_i, and its class label, ψ_i, we have used two class-specific attention layers to map it to the space of its won class and the entire data set of all classes.

(1) Class-Specific Attention Layer. To represent the ith sample, we pay attention to the samples of the same class, j: $\psi_i = \psi_j$. The attention weight is again calculated according to the similarity between u_i and u_j:

$$s\left(u_i, u_j; \iota_\psi\right) = \varphi\left(\iota_\psi^\top \begin{bmatrix} u_i \\ u_j \end{bmatrix}\right), \quad \forall j:\ \psi_i = \psi_j, \tag{15}$$

where the similarity function is based on the concatenation of u_i and u_j, a dense layer parameterized by ι_ψ, and an activation layer $\varphi(\cdot)$. The attention weights from the ith sample to class ψ are calculated by softmax normalization over the samples of the class ψ:

$$\delta_{ij}^\psi = \frac{\exp\left(s\left(u_i, u_j; \iota_\psi\right)\right)}{\sum_{j\prime:\ \psi_{j\prime} = \psi} \exp\left(s\left(u_i, u_{j\prime}; \iota_\psi\right)\right)}, \quad \forall j:\ \psi_j = \psi. \tag{16}$$

The class-specific attention representation of the ith sample regarding to class ψ is the combination of the weighted samples of class ψ:

$$P_i^{\psi} = \sum_{j:\ \psi j=\psi} \delta_{ij}^{\psi} \Omega_{\psi}^{\top} u j, \tag{17}$$

where Ω_{ψ} the projection matrix.

(2) Global Attention Layer. Beside the class-specific attention layers, we also build a global attention layer to represent the ith sample to all the data samples of the entire data set. The attention weights are calculated from the ith sample to all samples, $j \in X$. The similarity between the ith and jth samples is also based on a concatenation, a dense, and a activation layer:

$$s\left(u_i, u_j; \pi\right) = \phi\left(\pi^{\top}\begin{bmatrix} u_i \\ u_j \end{bmatrix}\right), \quad \forall j:\ j \in X, \tag{18}$$

where π is the dense layer parameter. Accordingly, the weights of attention are normalized by a softmax:

$$\varpi_{ij} = \frac{\exp\left(s\left(u_i, u_j; \pi\right)\right)}{\sum_{j_l:\ j_l \in X} \exp\left(s\left(u_i, u_{j_l}; \pi\right)\right)}, \quad \forall j \in X. \tag{19}$$

The global attention representation of the ith sample is

$$q_i = \sum_{j:\ j \in E} \varpi_{ij} \Upsilon^{\top} u_j, \tag{20}$$

where Υ the the projection matrix.

With these layers, for each data sample, we have two representations, which are class-specific attention vector, $p_i^{\psi i}$, and a global attention vector, q.

2.3. Network Training. There are many parameters of the proposed network. To learn these parameters, we firstly model a minimization problem with the training data set and then develop an iterative optimization algorithm to solve it.

2.3.1. Objective Function. To train the parameters of the cross-feature attention network and class-specific/global attention network, we consider the following two problems:

(1) Minimization of Within-Class Scattering. For each class ψ, we hope that its samples' representations of this class are not scattered so that they can be gathered as close as possible. The samples' class-specific representations are $p_i^{\psi}|_{i:\ \psi_i=\psi}$. To measure the within-class scattering, we first calculate the mean vector of this class as

$$\mu^{\psi} = \frac{1}{n_{\psi}} \sum_{i:\ \psi_i=\psi} p_i^{\psi}, \tag{21}$$

where n_{ψ} is the number of samples of the class ψ. The within-class scattering measure of class ψ is calculated as

$$S_{\psi}^{W} = \sum_{i:\ \psi_i=\psi} \mathrm{Tr}\left(\left(p_i^{\psi} - \mu_{\psi}\right)\left(p_i^{\psi} - \mu_{\psi}\right)\top\right), \tag{22}$$

where $\mathrm{Tr}(\cdot)$ is the trace of a matrix. The following minimization problem is modeled to optimize the parameters,

$$\min \sum_{\psi=1}^{C} S_{\psi}^{W}, \tag{23}$$

so that, for all the classes, the within-class scattering is minimized jointly.

(2) Maximization of Interclass Scattering. We also propose to maximize the scattering of different classes. For this purpose, we firstly calculate an overall mean vector over the entire data set, using the global attention representations:

$$\mu = \frac{1}{n} \sum_{i:\ i \in X} q_i. \tag{24}$$

Meanwhile, in the global attention representation space, we also calculate the mean vectors for each class, ψ:

$$\rho^{\psi} = \frac{1}{n\psi} \sum_{i:\ \psi i=\psi} q_i. \tag{25}$$

The interclass scattering is measured as

$$S_B = \sum_{\psi=1}^{C} n_{\psi} \mathrm{Tr}\left(\left(\rho^{\psi} - \mu\right)\left(\rho^{\psi} - \mu\right)^{\top}\right). \tag{26}$$

To separate different classes, we propose to maximize it:

$$\max S_B. \tag{27}$$

The overall objective of this problem is the combination of (23) and (27).

A minimization problem is proposed as follows to learn the parameters of the network:

$$\mathrm{Min}_{\Pi}\left\{o\left(\Pi\right) = \sum_{\psi=1}^{C} S_{\psi}^{W} - S_B + C\|\Pi\|_F^2 = \sum_{\psi=1}^{C} \sum_{i:\ \psi i=\psi} \mathrm{Tr}\left(\left(p_i^{\psi} - \mu_{\psi}\right)\left(p_i^{\psi} - \mu_{\psi}\right)^{\top}\right) - \sum_{\psi=1}^{C} \mathrm{Tr}\left(\left(p_i^{\psi} - \mu\right)\psi^{\top}\right) + C\|\Pi\|_F^2\right\}, \tag{28}$$

where $\Pi = \left\{\left(A_l, B_l,\ E_l, \Theta_l, \Phi_l, \Psi_l, W_l,\ V_l, R_l\right)|_{1=1}^{L},\ \left(\iota_{\psi}, \Omega_{\psi}\right)|_{\psi=1}^{C}, \pi, \Upsilon\right\}$, $\|\Pi\|_F^2$ is the squared ℓ_2 norms of the parameters to prevent the overfitting problem, and C is the weight of the squared ℓ_2 norm term.

2.3.2. Objective Optimization. To solve the problem of (28), we employ the algorithm of Adam [22]. This algorithm is based on gradient descent updating of the parameters, and its optimization is for the stochastic objective. The lower-order moments is updated adaptively.

2.4. Model Inference. With the trained parameters of the network, we can inference the class label of a test sample. Its survived feature vector is y, and its newly added feature vector is z. We firstly represent it by the trained cross-feature network as u, according to (14). Then, for each class, we calculate its class-specific representation p_ψ and a global representation q. Then, we calculate the distance between the specific representation of the test sample and the class mean regarding to the class ψ:

$$s_w(\psi) = \mathrm{Tr}\left(\left(p_\psi - \mu_\psi\right)\left(p_\psi - \mu_\psi\right)^\top\right). \tag{29}$$

Moreover, we also update the mean vector of this class in the global representation space by

$$\rho_\psi = \frac{1}{n\psi + 1}\left(q + \sum_{i:\ \psi i = \psi} q_i\right). \tag{30}$$

With the updated class mean, we recalculate its distance to the overall mean vector μ as follows:

$$s_b(\psi) = \mathrm{Tr}\left(\left(\rho_\psi - \mu\right)\left(\rho_\psi - \mu\right)^\top\right). \tag{31}$$

The overall score of assigning the test sample the class ψ is the difference of $s_b(\psi)$ and $s_w(\psi)$:

$$\eta(\psi) = s_b(\psi) - s_w(\psi). \tag{32}$$

It measures how close the test sample is to the class ψ and how it makes the class ψ far away from the other classes. The test sample is assigned to the class which gives the largest score:

$$\psi^* = \arg\max_{\psi=1}^{C} \eta(\psi). \tag{33}$$

3. Experiments

In this section, we evaluate the proposed method cross-feature attention network (CFAN) experimentally. We firstly introduce the data sets and the experimental setting, then give the experimental results, and summarize the observations from the results.

3.1. Data Sets and Experimental Protocol

3.1.1. Data Sets. In the experiment, we use four data sets as benchmarks, including two computer vision data sets, an Internet of things (IoT) data set, and a bio-informatics data set. The statistics of the data sets is given in Table 1. The details of the data sets are as follows.

(i) Satimage is an image data set. It has 6,431 images, and the problem is to categorize each image into one of the 7 categories, including red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, mixture class (all types present), and very damp grey soil. Each image is represented by 36 features, which are the 9 pixels in the neighborhood of 4 spectral bands [23].

(ii) MINIST is a hand-written digit data set. It has a training set of 60,000 images and a test set of 10,000 images. Each image has 28×28 pixels and a class label of ten digits; thus, it is a 10-class classification problem [24].

(iii) SensIT vehicle is an acoustic data set. It has 78,823 training samples and 19,705 test samples. For each sample, it has 50 features. The learning problem of this data set is a 3-class classification problem [25].

(iv) Protein data set is a bio-informatics data set. It has 32,661 training data samples, 6,621 test data samples, and 2,871 evaluation samples. Each sample has 357 features. The problem for this data set is a 3-class classification problem [26].

3.1.2. Protocol. To perform the experiments, we split the features set of each data set into three subsets, so that each set has an equal size. The three feature sets are the vanished features, survived features, and newly added features, respectively. To create the training and test data set, we use the 10-fold cross-validation protocol. A data set is split into 10 folds of equal sizes, and each fold is used as a test set, while the other 9 folds are combined to form a training set. We firstly train the model over the training set and then test it over the test set. To measure the accuracy of the classification, classification accuracy is used. It is calculated as the rate of correctly classified samples from the overall test samples.

3.2. Experimental Results. In this section, given the data and protocol, we perform the experiments and report the results. We evaluate the proposed method from three different aspects, including the sensitivity to the tradeoff parameter, the running time, and the performance compared to the state of the art.

3.2.1. Parameter Sensitivity. In the objective of our model in (28), there is a tradeoff parameter C to balance the regularization term and other terms. It controls the weight of the model complexity. To evaluate how it affects the performance of the model, we plot the curves of the accuracy against the changing values of C in Figure 2. From the figure, we observe that, with the increase of the weight of the regularization term, the accuracy is improved slightly. However, generally speaking, the performance keeps stable regarding the change of the value of C. A simpler model with a larger value of C can improve the quality of the model, but the improvement is not significant.

3.2.2. Running Time Analysis. The running time of the training process of the model is also studied in the

Table 1: Statistics of data sets.

Data set	# of data sample	# of class	# of feature
Satimage	6,431	7	36
MINIST	70,000	10	784
SensIT vehicle	98,528	3	50
Protein	42,153	3	357

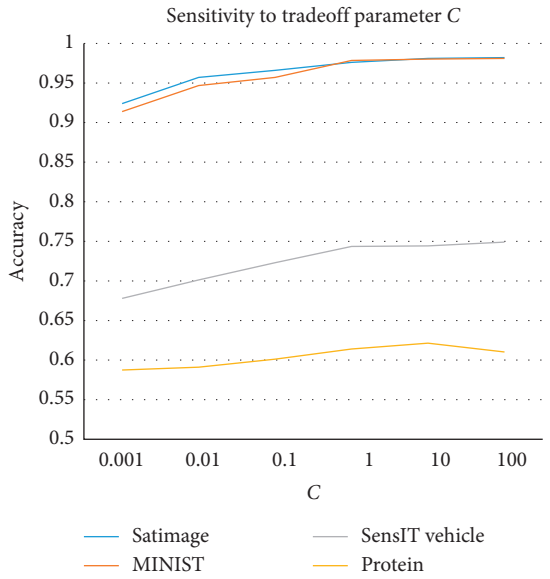

Figure 2: Sensitivity curve of tradeoff parameter C.

experiment. Moreover, the running time of the classification of the test samples is also reported. The running time over four benchmark data sets is given in Figure 3.

(1) From the figure, we can see that the training time is longer than the test time for each data set. This is natural because the training algorithm scans each training sample for many iterations, and the number of training samples is also larger than the test sample. Meanwhile, in the test process, each test sample is only scanned once.

(2) SensIT Vehicle and MINIST have a longer running time than that of Satimage and Protein since the data set sizes are larger.

(3) With ten thousands of samples, the running time of training and test is only hundreds of seconds. This indicates the efficiency of the algorithm.

3.2.3. Comparison to State of the Arts. We compared our algorithm CFAN against the other three state-of-the-art algorithms, including

(i) One-pass incremental and decremental learning approach (OPID) [6]

(ii) Heterogenous feature-based structural adaptive regression (HF-SAR) [7]

(iii) Online streaming feature selection (OSFS) [21]

The accuracy of these methods is reported in Figure 4. We have the following observations from this figure:

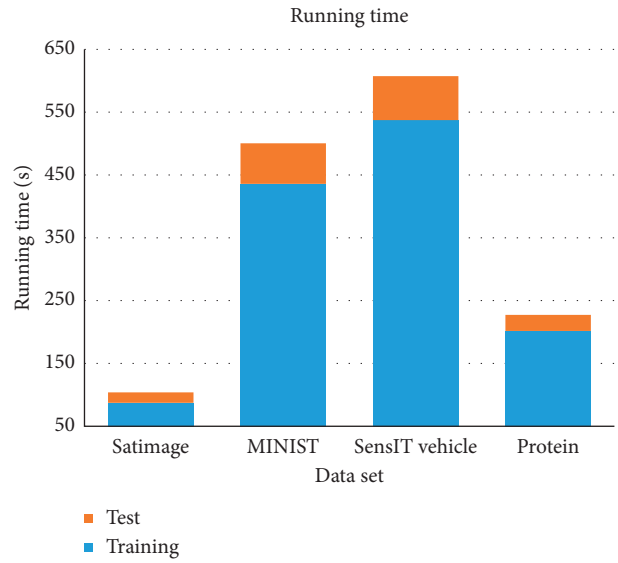

Figure 3: Running time of training and test process.

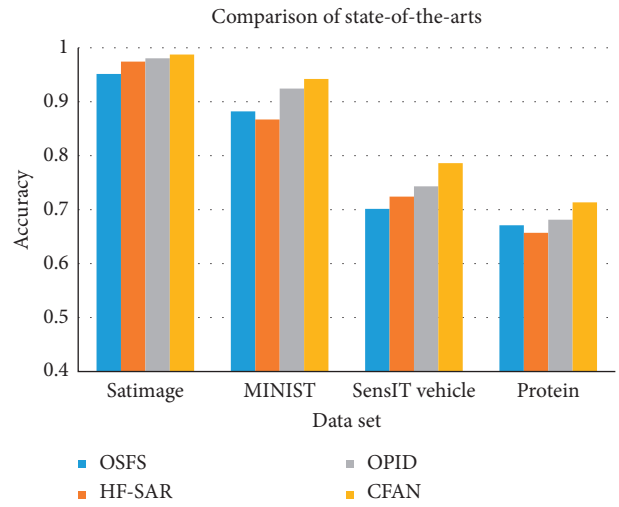

Figure 4: Comparison of accuracy of state of the arts.

(1) In all the cases, the CFAN algorithm keeps outperforming the compared methods, especially in the most challenging data sets, SensIT Vehicle and Protein. This is a strong indicator of the effectiveness and advantage of CFAN over the compared methods. The main reason is the power of the cross-feature attention layers which takes advantage of the essential nature of the changing features due to the evolving sensors.

(2) The second best algorithm is OPID, which is also specially designed for the IDF problem. However, because it used as a linear model to model the evolving features, it fails to capture the complex pattern of the features. In contrast, CFAN used the deep attention layers for this purpose, thus giving much better results.

(3) HF-SAR and OSFS give the worst results. They can handle some special cases of IDF but are not perfect solutions for this problem.

4. Conclusion

In this paper, we proposed a novel solution for the machine learning problem with evolving features. In the process of learning, old features vanished, while new features are argumented, and the remaining features survived. To handle the evolving features, we use a deep network structure with a newly designed cross-feature attention layer. This layer fills the gap between the survived features and vanished/argumented features, by paying attention from/to different feature sets. In this way, data samples with a different sets of features are mapped to a common feature space. To learn the attention network parameters, we proposed to construct the class-specific attention layer to minimize the within-class scattering and the global attention layer for the maximization of interclass scattering. Experimental results show the stability of the algorithm and the outperforming of the proposed algorithm against the other methods.

Data Availability

The dataset used in this paper are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Innovation Capability Improvement Plan of Heibei Province of People's Republic of China (program no. 20556103D) and the Science and Technology Research and Development Plan of Qinhuangdao City, People's Republic of China (program no. 202003B054).

References

[1] J. Hwang, C. Shin, and H. Yoe, "Study on an agricultural environment monitoring server system using wireless sensor networks," *Sensors*, vol. 10, no. 12, Article ID 11189, 2010.

[2] P. Jiang, H. Xia, Z. He, and Z. Wang, "Design of a water environment monitoring system based on wireless sensor networks," *Sensors*, vol. 9, no. 8, pp. 6411–6434, 2009.

[3] M. F. Othman and K. Shazali, "Wireless sensor network applications: a study in environment monitoring system," *Procedia Engineering*, vol. 41, pp. 1204–1210, 2012.

[4] G. Xu, W. Shen, and X. Wang, "Applications of wireless sensor networks in marine environment monitoring: a survey," *Sensors*, vol. 14, no. 9, Article ID 16932, 2014.

[5] J. Dong, Y. Cong, G. Sun, T. Zhang, and X Xu, "Evolving metric learning for incremental and decremental features," 2020, https://arxiv.org/abs/2006.15334.

[6] C. Hou and Z. H. Zhou, "One-pass learning with incremental and decremental features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 11, pp. 2776–2792, 2017.

[7] Z. Ma, Y. Yang, N. Sebe, and A. G. Hauptmann, "Knowledge adaptation with PartiallyShared features for event DetectionUsing few exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 9, pp. 1789–1802, 2014.

[8] H. V. Ribeiro, S. Mukherjee, and X. H. Zeng, "The advantage of playing home in nba: microscopic, team-specific and evolving features," *PLoS One*, vol. 11, no. 3, Article ID e0152440, 2016.

[9] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," 2017, https://arxiv.org/abs/1706.03762.

[10] X. Wang, X. He, Y. Cao, M. Liu, and T. S. Chua, "KGAT: knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 950–958, Anchorage, AK, USA, August 2019.

[11] Y. Yan, J. Qin, B. Ni et al., "Learning multi-attention context graph for group-based re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, 2020.

[12] Y. Zhang, Z. Guo, and W. Lu, "Attention guided graph convolutional networks for relation extraction," 2019, https://arxiv.org/abs/1906.07510.

[13] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proceedings of the International Conference on Machine Learning*, pp. 3734–3743, PMLR, Long Beach, CA, USA, June 2019.

[14] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018, https://arxiv.org/abs/1803.02155.

[15] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proceedings of International Conference on Machine Learning*, pp. 7354–7363, PMLR, Long Beach, CA, USA, June-2019.

[16] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10076–10085, Seattle, WA, USA, April 2020.

[17] Y. Hao, Y. Zhang, K. Liu et al., "An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 221–231, Vancouver, Canada, July 2017.

[18] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, "Cross attention network for few-shot classification," 2019, https://arxiv.org/abs/1910.07677.

[19] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: criss-cross attention for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 603–612, Seoul, South Korea, November 2019.

[20] K. H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proceedings of the European Conference on Computer Vision*, pp. 201–216, ECCV), Munich, Germany, September 2018.

[21] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1178–1192, 2012.

[22] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[23] Satimage, 2014,https://datahub.io/machine-learning/satimage.

[24] Mnist Digits Classification Dataset, 2014,https://keras.io/api/datasets/mnist/.

[25] M. F. Duarte and H. Y. Hen, "Vehicle classification in distributed sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 826–838, 2004.

[26] J. Y. Wang, "Application of support vector machines in bioinformatics," M.Sc. thesis, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2002.