This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

Deep Fuzzy Cognitive Maps for Interpretable Multivariate Time Series Prediction

Jingyuan Wang, Zhen Peng, Xiaoda Wang, Chao Li, and Junjie Wu

***Abstract*—The Fuzzy Cognitive Map (FCM) is a powerful model for system state prediction and interpretable knowledge representation. Recent years have witnessed the tremendous efforts devoted to enhancing the basic FCM, such as introducing temporal factors, uncertainty or fuzzy rules to improve interpretation, and introducing fuzzy neural networks or Wavelets to improve time series prediction. But how to achieve high-precision yet interpretable prediction in cross-domain real-life applications remains a great challenge. In this paper, we propose a novel FCM extension called *Deep FCM* for multivariate time series forecasting, in order to take both the advantage of FCM in interpretation and the advantage of deep neural networks in prediction. Specifically, to improve the predictive power, Deep FCM leverages a fully connected neural network to model connections (relationships) among concepts in a system, and a recurrent neural network to model unknown exogenous factors that have influences on system dynamics. Moreover, to foster model interpretability encumbered by the embedded deep structures, a partial derivative-based approach is proposed to measure the connection strengths between concepts in Deep FCM. An Alternate Function Gradient Descent algorithm is then proposed for parameter inference. The effectiveness of Deep FCM is validated over four publicly available datasets with the presence of seven baselines. Deep FCM indeed provides an important clue to building interpretable predictors for real-life applications.**

***Index Terms*—Fuzzy Cognitive Maps, Time Series Prediction, Deep Neural Networks, Interpretable Prediction**

I. Introduction

The Fuzzy Cognitive Map (FCM) is a flexible and powerful model for system state prediction and interpretable knowledge representation [1]. The FCM model describes a system with multiple interactive components (*i.e.*, concepts) as a weighted directed graph, where the vertexes denote system components and edges denote the interactions between components. Since the knowledge about a system is represented as a graph with clear interactive relationships, FCM is deemed naturally interpretable for system dynamics and has been widely adopted in many interpretation-sensitive prediction applications, such as public policy making [2], business management [3], healthcare diagnosis [4], and behavioral analysis [5].

J. Wang, is with School of Computer Science and Engineering, Beijing Advanced Innovation Center for Big Data and Brain Computing, and State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China.

Z. Peng is with School of Economics & Management, Beijing Institute of Petrochemical Technology, Beijing 102617, China.

X. Wang, and C. Li are with School of Computer Science and Engineering, and MOE Engineering Research Center of ACAT, Beihang University, Beijing 100191, China.

J. Wu is with School of Economics and Management, and Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China.

Corresponding author: Z. Peng, e-mail: zhenpeng@bipt.edu.cn

Interpretable knowledge representation and high performance prediction, unluckily, are often mutually exclusive. The FCM model is not an exception. In the basic FCM model, the graph edges can only describe static and linear relationships, which degrades the prediction performance of FCM in many complex real-world applications, especially when compared with neural network-based deep learning models [6]. The deep learning models, however, are often criticized for their black-box nature with incomprehensible variables in deep layers, let alone influence relationship explanations to the variables. How to improve the capability of FCM in non-linear dynamics prediction while keeping the interpretation advantage in the meanwhile, or in other words, to achieve satisfactory *interpretable prediction*, remains an open and essential problem.

In the literature, many extensions have been proposed to enhance the performance of the basic FCM model in terms of interpretation and prediction. For instance, temporal factors are introduced into the FCM framework to model dynamic relationships, resulting in Dynamical Cognitive Networks, Fuzzy Time Cognitive Maps and Evolutionary Fuzzy Cognitive Maps [7]–[9]. Fuzzy Grey Cognitive Maps, Intuitionistic Fuzzy Cognitive Maps, and Rough Cognitive Maps are proposed to model uncertain relationships among system components [10]–[12]. Rule-based Fuzzy Cognitive Maps and Extended Fuzzy Cognitive Maps adopt logic rules to express non-linear relationships in FCM [13], [14]. The fuzzy neural networks and wavelet transform are also adopted to improve the performance of the FCM framework in time series forecasting applications [15]–[18]. While the above studies indeed have improved the basic FCM model in different aspects, it is still in great need to design a new FCM model to gain high-precision yet interpretable prediction power for cross-domain real-life applications, where non-linear complex dynamics with unknown exogenous factors are commonly seen.

In this paper, the advantage of deep neural network models in high-performance prediction is introduced into the interpretable FCM framework to build a novel interpretable predictor called *Deep FCM* (or DFCM for short). Our model is designed for the task of multivariate time series forecasting. It extends the basic FCM to a general framework, which consists of a fully connected neural network to model non-linear and non-monotonic influences among system concepts, and a recurrent neural network (RNN) to model unknown exogenous factors that have latent influence on system dynamics. An Alternate Function Gradient Descent algorithm is then carefully designed for efficient parameter inference of Deep FCM with built-in deep neural networks. In this way, Deep FCM is equipped with much greater power than the

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

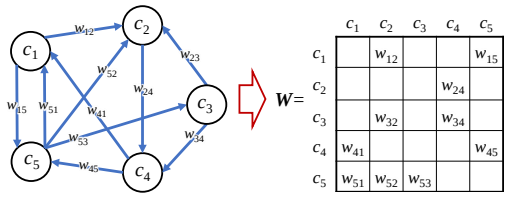


Fig. 1. An illustration of fuzzy cognitive maps.

basic FCM in time series prediction.

Beyond prediction, we also adopt a partial derivative-based method to measure the connection strength between each pair of system concepts. This is to ensure that the excellent interpretability of the basic FCM would not be undermined by the black-box nature of deep neural network components in Deep FCM. In this way, our model could achieve improved performance in multivariate time series prediction while keeping the interpretability of the FCM framework at the same time. That is why we called Deep FCM an *interpretable prediction* model.

The effectiveness of Deep FCM is verified over four publicly available datasets obtained from different application domains, and is compared with seven competitive baselines. The experimental results show that Deep FCM indeed can achieve much better performance in system state prediction. Meanwhile, the non-linear concept relationships in complex real-life systems indeed can be accurately captured and clearly interpreted by the partial derivative-based method. We also verifies the effectiveness of the RNN component in modelling periodical exogenous factors, which indeed improves the prediction power of Deep FCM.

II. Related Works

A. Fuzzy Cognitive Maps

In the basic FCM framework [1], a system consisting of several interactive components is described by three elements: *Concepts*, *Activation States*, and *Relationships*. Concepts represent components in a system, activation states represent states of components, and relationships represent influences among components.

As illustrated in Fig. 1, the FCM models a system with I concepts as a weighted directed graph. We denote the i-th graph vertex as c_i, which is used to express the i-th concept in the system. The edge weight for vertex i to vertex j is denoted as w_{ij}, which expresses the relationship of c_i to c_j. The value of w_{ij} is in the range of $[-1, 1]$. Moreover, each concept in FCM has a fuzzy activation state $a_i \in [0, 1]$. $a_i = 1$ means c_i is completely activated and $a_i = 0$ means completely unactivated. The activation states for c_i is a dynamic time series $\{a_i^{(1)}, \ldots, a_i^{(t)}, \ldots a_i^{(T)}\}$, where $a_i^{(t)}$ is the state at the time t. The state $a_i^{(t+1)}$ in FCM is influenced by the states of other concepts at the time t as

$$a_i^{(t+1)} = \varphi\left(a_i^{(t)} + \sum_{j \neq i} w_{ji} a_j^{(t)}\right), \tag{1}$$

where the function $\varphi(\cdot)$ is a membership function to fuzzify the activation states in $[0, 1]$, and the value of w_{ij} is in the range of $[-1, 1]$ [19].

In real-world applications, the activation levels $\{a_i^{(t)}\}$ are observable time series, and the relationships w_{ij} are unknown knowledge to be learnt from the observable activation levels. Given random initial values, the DHL algorithm adjusts w_{ij} using the observable data at time t as follows:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \lambda^{(t)}\left(\Delta a_i^{(t)} \Delta a_j^{(t)} - w_{ij}^{(t)}\right), \tag{2}$$

where $\Delta a_i^{(t)} = a_i^{(t)} - a_i^{(t-1)}$, and $\lambda^{(t)} = 0.1(1 - t/(1.1q))$ is a dynamic learning rate, with the parameter q adopted to ensure that $w_{ij} \in [-1, 1]$. The value of $w_{ij}^{(t+1)}$ is iteratively updated until convergence or some stopping criterion is met. The DHL algorithm has many improved versions, such as NHL (Nonlinear Hebbian learning) [20] and AHL (Active Hebbian learning) [21]. Moreover, evolutionary optimizations are also adopted to learn $\boldsymbol{W}$, such as the real-coded genetic algorithm (RCGA) [22] and the particle swarm optimization (PSO) [23].

B. Extensions of FCM

Despite the great success made, the basic FCM yet has some limitations. Firstly, relationships in many real-world systems are highly nonlinear and non-monotonic; however, the relationships modeled by basic FCM are linear and monotonic. In the literatures, many FCM extensions have been proposed to overcome this drawback [24]. One main stream of these extensions is using non-linear tools, such as logic rules, to describe complex concept relationships. For instance, RBFCM uses qualitative fuzzy rules to replace the quantitative mathematical description of relationships [13], and FRI-FCM uses fuzzy IF-THEN rules to express non-linear relationships [25], [26]. The other stream of the extensions is to introduce uncertainty into relationships. For instance, FGCM uses the grey system theory to handle highly uncertain relationships in incomplete and small datasets [10], iFCM introduces the intuitionistic fuzzy sets to handle the hesitancy in human decision makings [11], BDD-FCM replaces absolute linguistic terms as belief degree distributions to describe uncertain relationships [27], and the rough sets are introduced by RCM to represent diversity of the relationship among concepts [12].

The second limitation of the basic FCM is its relationships are static, which hinders its applications in dynamic systems. To overcome this drawback, temporal factors are introduced into FCM by many studies. For instance, DCN introduces a dynamic function into FCM relationships [7], [28], DRFCM adopts a reinforced learning procedure to update relationships dynamically [29], EFCM proposes asynchronous updates of the variables to handle dynamics of concept interactions [9], and TAFCM uses timed automata to model dynamic relationships between concepts [30]. A common idea of these works is to model relationships as a function of time.

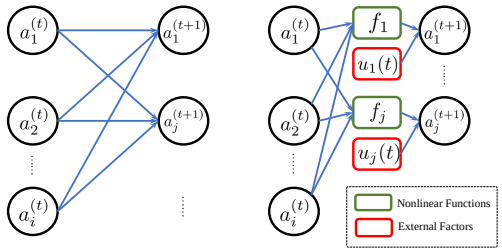
In summary, while the above-mentioned studies perform excellently in adapting FCM to nonlinear and dynamic relationships, a drawback of these extension models is users have to design complicated logic rules, uncertain and dynamic

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

TABLE I
MATH NOTATIONS.

Notation	Definition
c_i	The i-th concept in a system.
a_i	The fuzzy activation state of the concept i. The instance at time t is denoted as $a_i^{(t)}$.
$\boldsymbol{a}$	The fuzzy activation state vector of all concepts (the system state). The instance at time t is denoted as $\boldsymbol{a}^{(t)}$.
w_{ij}	The relationship of c_i to c_j, which is a constant in basic FCM but is a function of $\boldsymbol{a}$ in DFCM.
$f_i(\boldsymbol{a})$	The function to model the relationship of the system state $\boldsymbol{a}$ to a_i, named as f-function.
$u_i(t)$	The function to model the influence of exogenous factors to a_i, named as u-function.
$y_{(m,k)}$	The output of the m-th neuron in the k-th hidden layer of a f-function. $y_{(m,k)}^{(t)}$ is an instance at time t
$v_{(nm,k)}$	The weight of the n-th input of the m-th neuron in the k-th hidden layer of the f-function.
r_{ij}	The general relationship of the concept c_i to c_j, which is a function of $\boldsymbol{a}$ in DFCM.
$w_{ij}(a_k)$	A general strength measurement of the causal relationship of the concept c_i to c_j.



(*a*) Basic FCM (*b*) Deep FCM

Fig. 2. The framework of Deep FCM.

function according to specific applications. On the contrary, the representational learning capacity of neural networks lets users free from complicated rules designing. There is much room in improving the adaptability of the FCM framework in different application scenarios. One possible way is to leverage the representational learning ability of neural networks.

C. FCM for Time Series Predictions

The capabilities of FCM in time series modeling have already been widely acknowledged. Ref. [31] proposed a framework that first transforms state of a univariate time series into fuzzy sets and then uses a basic FCM model to predict the time series. Ref. [32] uses historical states in a moving window as concepts of basic FCM models to predict future state of a univariate time series. Ref. [32] proposed a mechanism to optimize the FCM structure, membership functions and moving window size in time series prediction dynamically. Ref. [33] improved the framework of [31] by using fuzzy C-means to transform time series into information granules. Ref. [34] adopted the ARIMA model to improve the performance of FCMs in time series prediction. In order to handle large-scale nonstationary time series, Wavelet-HFCM [15] applies redundant Haar wavelet transform to decompose univariate time series into multivariate time series, and uses ridge regression to train FCM models for forecasting.

The FCM model was also applied in the multivariate time series prediction problem, where the time series of multivariate are considered as states of concepts in a system [16], [35], [36]. Papageorgiou *et al.* [35] proposed a modified error function to optimize the performance for multi-step multivariate time series prediction of FCM. Froelich *et al.* [36] proposed a dynamic optimization for FCM parameter and structure selection in multivariate time series prediction. Papageorgiou *et al.* [16] proposed a two-stage prediction model which uses evolutionary FCMs to select the most important attributes as inputs in an ANN to make time series prediction.

Neural network structures were also adopted by FCM for time series prediction. Ref. [17] implements FCM based on a fuzzy neural network for time series prediction, and Ref. [18] adopts a similar FCM structure to model chaotic time series. However, in order to infer and express relationships between concepts, the structures of fuzzy neural networks in [17] and [18] are strictly limited with small numbers of layers.

III. DEEP FUZZY COGNITIVE MAPS

In this section, we propose the *Deep Fuzzy Cognitive Maps* model (deep FCM or DFCM for short) for multivariate state time series prediction and influence analysis among systematic concepts. Fig. 2 shows the framework of Deep FCM. The notations used in Deep FCM are listed in Table I.

A. Time Series Fuzzification

Given a system consisting of a group of concepts, we denote the original time series of a concept j as $\boldsymbol{x}_j = \left(x_j^{(1)}, \ldots, x_j^{(t)}, \ldots, x_j^{(T)}\right)^\top$, $x_j^{(t)} \in \mathbb{R}$, $\forall\, j, t$. The deep FCM first adopts a z-score normalization to preprocess the raw state value $x_j^{(t)}$ as

$$z_j^{(t)} = \frac{x_j^{(t)} - \mu_j}{\sigma_j}, \tag{3}$$

where μ_j and σ_j are the mean and standard deviation of $\boldsymbol{x}_j$. Next, deep FCM uses a sigmoid membership function to fuzzify the normalized time series $z_j^{(t)}$ into $a_j^{(t)} \in (0, 1)$ as follows:

$$a_j^{(t)} = \varphi\left(z_j^{(t)}\right) = \frac{1}{1 + e^{-z_j^{(t)}}}, \tag{4}$$

where $\varphi(\cdot)$ is the sigmoid membership function that has a range of $(0, 1)$. Apparently, when $z_j^{(t)} = +\infty$, $a_j^{(t)} = 1$

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

indicating the *active* state, and when $z_j^{(t)} = -\infty$, $a_j^{(t)} = 0$ indicating the *inactive* state. When $z_j^{(t)} \in (-\infty, +\infty)$, $a_j^{(t)} \in (0, 1)$, which indicates the state of *active to a certain degree*. In this way, the original time series values of concepts are represented by the fuzzy values of activation levels in [0,1].

Given a fuzzy activation state $a_j^{(t+1)} \in [0, 1]$ predicted by DFCM, we use the following function to defuzzify a fuzzy activation state as its raw value c_j:

$$x_j^{(t+1)} = \varphi^{-1}\left(a_j^{(t)}\right) \cdot \sigma_j + \mu_j. \tag{5}$$

In the default assumption, input time series $\boldsymbol{x}_j$ are not in crisp values. For the condition $x_j^{(t)} \in \{0, 1\}$, we skip the normalization and fuzzification steps and directly set $a_j^{(t)} = x_j^{(t)}$. When $a_j^{(t)} = 1$ indicating the *active* state, and when $a_j^{(t)} = 0$ indicating the *inactive* state.

B. Modelling Nonlinear Influence

One drawback of the basic FCM is its weak capacity in modeling nonlinear relationships. To deal with this, Deep FCM extends Eq. (1) of the basic FCM to a general form as

$$a_j^{(t+1)} = \varphi\left(u_j(t) + f_j\left(\boldsymbol{a}^{(t)}\right)\right). \tag{6}$$

Here, the function $f_j(\cdot)$ is used to model relationships of $\boldsymbol{a}$ to a_j, where $\boldsymbol{a}^{(t)} = (a_1^{(t)}, \ldots, a_i^{(t)}, \ldots, a_I^{(t)})^\top$ denotes the activation states of all concepts (*i.e.*, the system state) at time t. The function $u_j(t)$ is used to model influences of unknown exogenous factors to a_j. We name the two functions as the f-function and u-function, respectively. In Eq. 6, the summation of the f-function and the u-function is fuzzified by a sigmoid membership function φ to generate $a_j^{(t+1)}$. Obviously, when $u_j(t) = 0$ and $f_j(\boldsymbol{a}^{(t)}) = \sum_{i=1}^{I} w_{ij} a_i^{(t)}$ with $w_{jj} = 1$, DFCM degenerates to the basic FCM. In other words, the basic FCM is a special case of DFCM.

The neural network is a powerful model with universal approximation capability [37]. The f-functions of DFCM are implemented by feedforward neural networks [6]. Specifically, we define $f_j(\boldsymbol{a}^{(t)})$ in Eq. (6) as a feedforward neural network with K hidden layers. The number of neurons in the layer k is denoted by M_k. In the time slice t, the output of the m-th neuron in the k-th layer, *i.e.,* $y_{(m,k)}^{(t)}$, is generated by

$$y_{(m,k)}^{(t)} = \mathrm{ReLU}\left(\sum_{n=1}^{M_{k-1}} v_{(nm,k)} y_{(n,k-1)}^{(t)}\right), \tag{7}$$

where $v_{(nm,k)}$ is the connection weight from the neuron n in the layer $k-1$ to the neuron m in the layer k. $\mathrm{ReLU}(\cdot)$ is a Rectified Linear Unit (ReLU) activation function, which is defined as

$$\mathrm{ReLU}(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases}. \tag{8}$$

As mentoined in Ref. [6] the ReLU activation function have advanced performance. Moreover, ReLU can also ensure $f_j(\boldsymbol{a}) = 0$ at the origin $a_1 = \cdots = a_i = \cdots = a_I = 0$, which is consistent with the basic FCM.

In the input layer of $f_j(\boldsymbol{a}^{(t)})$, we set $y_{(n,0)}^{(t)} = a_n^{(t)}$. In the output layer, we calculate a predictive output as

$$y_{(K+1)}^{(t+1)} = \sum_{n=1}^{M_K} v_{(n1,K+1)} y_{(n,K)}^{(t)} = f_j(\boldsymbol{a}^{(t)}). \tag{9}$$

In the f-function, we do not include a bias term in neurons and do not applied the ReLU activation to the output layer neither. Both of the two treatments are to ensure that DFCM, a deep-structure enhanced FCM, is consistent with the basic FCM. As shown in Eq. (1), it is obvious that the expression of the basic FCM does not contain bias terms, which allows $a_i^{(t+1)} = 0$ at the origin $a_1 = \cdots = a_i = \cdots = a_I = 0$. In order to ensure that DFCM has the same feature, we did not include bias terms in DFCM, which implies that $f_j(\boldsymbol{a}) = 0$ at the origin $a_1 = \cdots = a_I = 0$. In addition, it is easy to note that the term $a_i^{(t)} + \sum_{j \neq i} w_{ji} a_j^{(t)}$ is in the range of $(-\infty, \infty)$. Therefore, in order to ensure $f_j(\boldsymbol{a}) \in (-\infty, \infty)$, we did not apply the ReLU activation to the output layer, whose output would be in the range of $[0, \infty)$. Given the above treatments, it is obvious that when $u_i(t) = 0$ and $f_i(\boldsymbol{a}^{(t)}) = \sum_{i=1}^{I} w_{ji} a_j^{(t)}$ with $w_{jj} = 1$, the expression of DFCM in Eq. (6) degenerates to the form of the basic FCM in Eq. (1).

C. Modelling Exogenous Factors

The exogenous factors in the deep FCM refer to those *exogenous* factors that have influence to the system state $\boldsymbol{a}$ but can not be predefined and directly measured. Let us take for example the Deep FCM for a road transportation system (more details can be found in the experimental section). In this case, the road segments can be modeled as concepts and whether the segment congest can be modeled as activation states. The traffic speeds of near road segments can influence each other, which form relationships among concepts and can be modeled by f-functions. However, the traffic speeds are also influenced by some exogenous factors, such as the commuting patterns of residents, traffic controls, important events and so on. Because the states of these exogenous factors cannot be directly measured, we cannot use the predefined FCM concepts to describe them. How to handle these factors is an age-old challenge for system modelling and attribution analysis studies [38]. In the literature of FCM, influences of exogenous factors are often modeled as static constant inputs [39], [40]. Our DFCM model is designed for time series prediction tasks, so we pay special attention to time-related factors and introduce a LSTM-based u-function to capture the exogenous factors with time dependence.

In the deep FCM framework defined in Eq. (6), the influence of exogenous factors to a_j are indirectly measured by a u-function as

$$u_j(t) = \varphi^{-1}\left(a_j^{(t+1)}\right) - f_j\left(\boldsymbol{a}^{(t)}\right), \tag{10}$$

i.e., the component of a_j that cannot be modeled by the system internal relationships through the f-function. The deep FCM adopts a Recurrent Neural Network (RNN) to implement the u-function as

$$u_j(t) = \mathrm{RNN}\left(t, \mathrm{mod}(t, \tau), u_j(t-1)\right), \tag{11}$$

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

which has three inputs: the time stamp t, the time stamp t modulo a period length τ, and the history state $u_j(t-1)$. $\mathrm{mod}(a,b)$ is a function to calculate a modulo b.

We design the u-function in the form of Eq. (11) based on three considerations: i) $u_j(t)$ is a function of time stamp t since the influence of exogenous factors usually change with time. ii) In many scenarios, exogenous factors exhibit periodicity, such as one day, one week, one month and the like, so the time stamp t modulo a period length τ is also adopted as an input. iii) Moreover, the dynamics of exogenous factors usually have "memory", *i.e.,* depend on their historical states. Therefore, we use Recurrent Neural Networks (RNN) to implement the u-function where the historical state $u_j(t-1)$ is adopted as an input. In practice, the version of RNN in DFCM is Long Short-Term Memory (LSTM) [41]. The calculation of $u_j(t)$ starts from $t=2$, where the input $u_j(t-1)$ is set as $u_j(1)=\varphi^{-1}(a_j^{(2)})-f_j(\boldsymbol{a}^{(1)})$.

D. Measuring Concept Relationships

The biggest advantage of FCM lies in its ability in uncovering concept relationships in a complex system. This advantage is also called as *interpretability* of FCM. The basic FCM uses w_{ij} to measure the strength of relationships between concepts. The value of w_{ij} has the following interpretations:

- $w_{ij}=0$ means concept c_i has no influence at all to concept c_j;
- $w_{ij}=(0,1]$ means c_i has a positive influence to c_j;
- $w_{ij}=[-1,0)$ means c_i has a negative influence to c_j.

From a computational perspective, the basic FCM also regards w_{ij} as "the level of a_j's increase when a_i increases". Analogously, we proposed a partial derivative-based method to measure the relationship of a_i to a_j in DFCM as:

$$r_{ij}(\boldsymbol{a})=\lim_{\Delta a_i\to 0}\frac{f_j(a_i,\boldsymbol{a}_{\neg i})-f_j(a_i+\Delta a_i,\boldsymbol{a}_{\neg i})}{a_i-(a_i+\Delta a_i)}=\frac{\partial f_j(\boldsymbol{a})}{\partial a_i}, \tag{12}$$

where $\boldsymbol{a}_{\neg i}$ denotes all the elements of $\boldsymbol{a}$ except a_i. The function $r_{ij}(\boldsymbol{a})$ expresses the degree of f_j's increasing when a_i increases Δa_i, given the system state $\boldsymbol{a}$ as a condition. To understand the necessity of introducing the condition, we can think about how human's body weight influences his health with a given age — ideal body weights are different for different ages.

Note that the function $r_{ij}(a_k)$ here is also a function of the activation states of unconcerned concepts: $\boldsymbol{a}_{\neg k}$. To remove the impact of the unconcerned concepts, we calculate the expectation of $r_{ij}(a_k)$ for all possible values of $\boldsymbol{a}_{\neg k}$ as

$$\bar{r}_{ij}(a_k)=\int_{D_{\boldsymbol{a}_{\neg k}}}P(\boldsymbol{a}_{\neg k})r_{ij}(a_k,\boldsymbol{a}_{\neg k})\ \mathrm{d}_\sigma, \tag{13}$$

where $P(\boldsymbol{a}_{\neg k})$ is the probability density function of $\boldsymbol{a}_{\neg k}$, and $\int_{D_{\boldsymbol{a}_{\neg k}}}\cdot\ \mathrm{d}_\sigma$ is an integral over all possible value of $\boldsymbol{a}_{\neg k}$. Furthermore, the overall influence of a_i to a_j is calculated as the expectation of r_{ij} for all possible values of $\boldsymbol{a}$, *i.e.,*

$$\bar{r}_{ij}^{(O)}=\int_{D_{\boldsymbol{a}}}P(\boldsymbol{a})r_{ij}(\boldsymbol{a})\ \mathrm{d}_\sigma. \tag{14}$$

In practices, $P(\boldsymbol{a})$ and $P(\boldsymbol{a}_{\neg k})$ are unknown. A practicable method is to use frequency to approximate probability. For a_k in a small interval $[\alpha,\beta]$, we assume there are M samples falling in this range. According to the Large Number Law, the $\bar{r}_{ij}$ for $a_k\in[\alpha,\beta]$ can be calculated approximately as

$$\bar{r}_{ij}(a_k)=\frac{1}{M}\sum_{a_k\in[\alpha,\beta]}r_{ij}\left(a_k,\boldsymbol{a}_{\neg k}\right). \tag{15}$$

The overall influence $\bar{r}_{ij}^{(O)}$ can then be approximated as

$$\bar{r}_{ij}^{(O)}=\frac{1}{M}\sum_{m=1}^{M}r_{ij}\left(\boldsymbol{a}^{(m)}\right), \tag{16}$$

where $\boldsymbol{a}^{(m)}$ is the system state of the m-th sample.

The FCM framework requires the values of relationships to be in the range of $[-1,1]$, so we use the hyperbolic tangent function to resize $\bar{r}_{ij}$ as

$$w_{ij}^{(C)}(a_k)=\mathrm{Tanh}\left(\bar{r}_{ij}(a_k)\right),\quad w_{ij}^{(O)}=\mathrm{Tanh}\left(\bar{r}_{ij}^{(O)}\right), \tag{17}$$

where $\mathrm{Tanh}(\cdot)$ is in the form of

$$\mathrm{Tanh}(z)=\frac{e^z-e^{-z}}{e^z+e^{-z}}. \tag{18}$$

The function $w_{ij}^{(C)}(a_k)$ is called the *Conditional Relationship Strength* of w_{ij} *w.r.t.* a_k, which is used to express how the relationship of c_i to c_j changes with the system state a_k. The variable $w_{ij}^{(O)}$ is called the *Overall Relationship Strength* of c_i to c_j, which is used to express the overall relationship strength of c_i to c_j.

The problem remains unsolved is to calculate the partial derivative $\partial f_j(\boldsymbol{a})/\partial a_i$ in Eq. (12) given a deep structure. According to the chain rule, the partial derivative of f_j to the input $y_{(m,k)}$ in the layer k can be recursively expressed as

$$\frac{\partial f_j}{\partial y_{(m,k)}}=\sum_n\frac{\partial f_j}{\partial y_{(n,k+1)}}\frac{\partial y_{(n,k+1)}}{\partial y_{(m,k)}}. \tag{19}$$

Based on the definition in Eq. (7), the partial derivative $\partial y_{(n,k+1)}/\partial y_{(m,k)}$ is calculated as

$$\frac{\partial y_{(n,k+1)}}{\partial y_{(m,k)}}=v_{(mn,k+1)}\varphi'\left(\sum_{m=1}^{M_{k+1}}v_{(mn,k+1)}y_{(m,k)}\right), \tag{20}$$

where φ' is the derivative of the ReLU activation function. Note that in the input layer, $a_i=y_{(i,0)}$. In this way, r_{ij} can be recursively calculated for any given system state $\boldsymbol{a}$.

IV. Parameters Inference

A. Objective Function

Letting $z_j^{(t)}=(x_j^{(t)}-\mu_j)/\sigma_j$, *i.e.*, the Z-score of $x_j^{(t)}$, our Deep FCM defined in Eq. (6) can be rewritten as

$$\hat{z}_j^{(t)}=f_j(\boldsymbol{a}^{(t)}|\theta_f)+u_j(t|\theta_u), \tag{21}$$

where $\hat{z}_j^{(t)}$ denotes the predicted value of $z_j^{(t)}$, and θ_f,θ_u denote the parameters of f_j and u_j, respectively.

Because $\hat{z}_j^{(t)}$ contains the influence of all concepts and exogenous factors, the residual between $\hat{z}_j^{(t)}$ and $z_j^{(t)}$ should

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

be a random error. We assume $e_j = \hat{z}_j^{(t)} - z_j^{(t)}$ is a zero mean Gaussian noise as

$$e_j \sim \mathcal{N}(0, \sigma_e^2), \quad \forall j, \tag{22}$$

where σ_e^2 is the variance. Given a training data set $Z_j = \{z_j^{(1)}, \ldots, z_j^{(t)}, \ldots, z_j^{(T)}\}$, the likelihood of Z_j for given the DFCM parameters θ_f, θ_u is expressed as

$$\begin{aligned} P(Z_j|\theta_f, \theta_u) &= \prod_{t=1}^{T} \mathcal{N}(\hat{z}_j^{(t)}, \sigma_e^2) \\ &= \prod_{t=1}^{T} \frac{1}{\sigma_e\sqrt{2\pi}} \exp\left(-\frac{(z_j^{(t)} - \hat{z}_j^{(t)})^2}{2\sigma_e^2}\right). \end{aligned} \tag{23}$$

The negative log likelihood of Z_j can be formulated as

$$-\ln P(Z_j|\theta_f, \theta_u, \sigma_e^2) \propto \sum_{t=1}^{T} \left(z_j^{(t)} - \hat{z}_j^{(t)}\right)^2. \tag{24}$$

We uses a the Maximum Likelihood Estimation (MLE) method to infer the parameters θ_f and θ_u, which is equal to minimize the loss function defined as

$$L(\theta_f, \theta_u) = \frac{1}{2} \sum_{t=1}^{T} \left(f_j(\boldsymbol{a}^{(t)}|\theta_f) + u_j(t|\theta_u) - z_j^{(t)}\right)^2. \tag{25}$$

B. Principle of AFGD Algorithm

The biggest difference between DFCM and the basic FCM lies in that the DFCM contains many deep neural network components, such as f-function and u-function. However, the traditional FCM training algorithms, including the Hebbian-like methods and the evolutionary optimizations, cannot be directly used to train deep neural networks, which motivates us to find a new training method, named Alternate Function Gradient Descent (AFGD), to optimize the objective in Eq. (25). Since the Back-Propagation (BP) algorithm [6] is an effective algorithm for deep neural network training, we designed the AFGD algorithm based on BP to train the DFCM model.

AFGD learns the parameters θ_f and θ_u via an iterative approach. Specifically, at the q-th round of iteration, AFGD updates θ_f and θ_u such that the following equations hold:

$$\begin{aligned} f_j^{(t)}\left(\theta_f^{(q)}\right) &= f_j^{(t)}\left(\theta_f^{(q-1)}\right) - \eta_f \cdot \left.\frac{\partial L(f_j)}{\partial f_j}\right|_{f_j = f_j^{(t)}\left(\theta_f^{(q-1)}\right)}, \\ u_j^{(t)}\left(\theta_u^{(q)}\right) &= u_j^{(t)}\left(\theta_u^{(q-1)}\right) - \eta_u \cdot \left.\frac{\partial L(u_j)}{\partial u_j}\right|_{f_j = f_j^{(t)}\left(\theta_u^{(q-1)}\right)}, \end{aligned} \tag{26}$$

where we denote $f_j^{(t)}(\theta_f^{(q)})$ as $f_j(\boldsymbol{a}^{(t)}|\theta_f^{(q)})$ and $u_j^{(t)}(\theta_u^{(q)})$ as $u_j(t|\theta_u^{(q)})$ for short, $\theta_f^{(q)}$ and $\theta_u^{(q)}$ are the parameters learnt in the q-th iteration, and η_f, η_u are two updating parameters representing the learning rates. Eq. (26) ensures that the parameters updating direction is along the negative gradient direction of the loss function to the functions f_j and u_j, so we name our algorithm as *Alternate Function Gradient Descent*.

According to the derivations in Appendix A, Eq. (26) is equal to following parameter iteration functions:

$$f_j(\boldsymbol{a}^{(t)}|\theta_f^{(q)}) = z_j^{(t)} - u_j(t|\theta_u^{(q-1)}), \tag{27}$$

$$u_j(t|\theta_u^{(q)}) = z_j^{(t)} - f_j(\boldsymbol{a}^{(t)}|\theta_f^{(q)}). \tag{28}$$

Algorithm 1 The AFGD algorithm.

1: **Input:** Training dataset $D = \{(t, \boldsymbol{a}^{(t)}, z_j^{(t)})\}_{t=1}^T$.
2: Initialize θ_f, θ_u randomly.
3: Initialize $y_j^{(t)} \leftarrow z_j^{(t)}$ for $t \in [1, T]$.
4: **repeat**
5: $\quad \theta_f \leftarrow \mathrm{BP}\left(f_j(\theta_f), \{(\boldsymbol{a}^{(t)}, y_j^{(t)})\}_{t=1}^T\right)$
6: $\quad y_j^{(t)} \leftarrow z_j^{(t)} - f_j(\boldsymbol{a}^{(t)}|\theta_f)$ for $t \in [1, T]$
7: $\quad \theta_u \leftarrow \mathrm{BP}\left(u_j(\theta_u), \{(t, y_j^{(t)})\}_{t=1}^T\right)$
8: $\quad y_j^{(t)} \leftarrow z_j^{(t)} - u_j(t|\theta_u)$ for $t \in [1, T]$
9: **until** convergence
10: **return** θ_f,θ_u

Algorithm 2 The BP algorithm.

1: **Input:** $\mathrm{BP}\left(g(\theta), \{(x^{(t)}, y^{(t)})\}_{t=1}^T\right)$, where $g(\theta)$ is a neural network, and $D = \{(x^{(t)}, y^{(t)})\}_{t=1}^T$ is a training dataset.
2: **repeat**
3: $\quad$ **for all** $(x^{(t)}, y^{(t)}) \in D$ **do**
4: $\quad\quad \mathrm{Loss}(\theta) = \left(y^{(t)} - g(x^{(t)}, \theta)\right)^2$
5: $\quad\quad \theta \leftarrow \theta - \lambda \dfrac{\partial \mathrm{Loss}(\theta)}{\partial \theta}$
6: $\quad$ **end for**
7: **until** convergence
8: **return** θ

Eqs. (27) and (28) could be intuitively understood as that the u-function and f-function alternately use the other's prediction residuals to train their parameters, that is why we call our algorithm as *Alternate* Function Gradient Descent. The prediction residuals of the f-function are the influences that cannot be modeled by the internal FCM concepts, *i.e.*, the influences of exogenous-factors, and therefore should be absorbed by the u-function. In contrast, the prediction residuals of the u-function correspond to removing the influences of exogenous factors from z_j, which should be modeled by the f-function.

C. Implementation of AFGD Algorithm

Alg. 1 and Alg. 2 give the pseudo-codes about the training algorithm. Alg. 1 establishes a framework of the AFGD algorithm. The main body of the algorithm is an iteration loop, *i.e.,* line 4 to line 9. In each loop, we alternately use the Back-propagation (BP) algorithm [42] to train the neural network functions $f_j(\theta_f)$ and $u_j(\theta_u)$ with the training set $\{(\boldsymbol{a}^{(t)}, y_j^{(t)})\}_{t=1}^T$ and $\{(t, y_j^{(t)})\}_{t=1}^T$, respectively. For each training sample, $\boldsymbol{a}^{(t)}$ and t are features and $y_j^{(t)}$ is a label. In line 6, the label $y_j^{(t)}$ is set as $y_j^{(t)} \leftarrow z_j^{(t)} - f_j(\boldsymbol{a}^{(t)}|\theta_f)$ for $u_j(\theta_u)$ training, and in line 8, $y_j^{(t)}$ is set as $y_j^{(t)} \leftarrow z_j^{(t)} - u_j(t|\theta_u)$ for the training of $f_j(\theta_f)$.

Alg. 2 gives the pseudocodes of the BP algorithm [42]. The main body of the BP algorithm is an iterative gradient descent loop, *i.e.,* from line 2 to line 7. In each loop, the algorithm traverses all training samples $(x^{(t)}, y^{(t)})$, and calculates a function $\mathrm{Loss}(\theta) = \left(y^{(t)} - g(x^{(t)}, \theta)\right)^2$ using $(x^{(t)}, y^{(t)})$. Here, $g(x^{(t)}, \theta) = f_j(\boldsymbol{a}^{(t)}, \theta_f)$ is for the f-function, and $g(x^{(t)}, \theta) = u_j(t, \theta_u)$ is for the u-function. The algorithm uses the negative gradient $-\lambda \partial \mathrm{Loss}(\theta)/\partial\theta$ to update the neural network parameters θ, where λ is a preset learning rate. The

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

TABLE II
STATISTICS OF THE FOUR DATASETS.

Statistics	AQI	Traffic	EPC.	Temp.
# of features	9	6	7	24
# of labels	5	6	3	2
# of records	8,783	3,602	21,899	2,763
Period	1 hour	10 min	1 min	15 min
Range	1 year	1 month	15 days	1 month

negative gradients for the neural networks f_j and u_j can be calculated using the chain rule [6].

V. EXPERIMENTAL RESULTS

In this section, we first compare multivariate time series prediction performance of DFCM with baselines over four datasets, and give some detail performance and interpretation analysis over two of the datasets.

A. Experimental Setup

1) Datasets: To estimate performance of the proposed deep FCM model, we adopted following four real-world multivariate time series datasets in our experiments:

• *AQI (Air Quality Indexes)*: This dataset contains time series of meteorological and air quality indexes that were collected from February 2017 to February 2018 in Beijing, China [1]. We use four meteorological indexes and five air quality indexes at the time t as inputs to predict the air quality indexes (AQIs) at the time $t+1$.

• *Traffic*: This dataset contains traffic speeds that were collected from 1st to 30th April in 2016 of six road segments in Beijing [2]. We use the traffic speeds of all segments at t as inputs to predict the speed of each road segment at $t+1$.

• *Power (Electric Power Consumption)*: This dataset contains measurements of electric power consumption in one household [3]. Different electrical quantities and some sub-metering values are available. The date used in our experiment were collected from 16th to 31st in December 2016. We use four electrical quantities and three sub-metering values as inputs to prediction the three sub-metering value at the next period.

• *Temperatures*: This dataset contains 24 features that are collected from a monitor system mounted in a domestic house [4]. The date used in our experiment were collected from 3rd March to 11th April in 2012. We use all features at time t to predict the indoor temperature of dinning-room and room at time $t+1$.

Table II summaries the statistics of the datasets. All of the datasets are publicly available.

[1]https://www.kdd.org/kdd2018/kdd-cup

[2]https://github.com/BuaaPercy/Traffic-DataSet-of-Beijing-Road

[3]http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption

[4]https://archive.ics.uci.edu/ml/datasets/SML2010

2) Baselines: We consider the following methods as baselines to compare:

• *FCM* [43]: This baseline treats multivariate time series as a multip-concept sytem and uses a basic FCM whose weights are trained by a Real-Coded Genetic Algorithm (RCGA) to predict concept states of the system.

• *ANN* [44]: This baseline uses artificial neural networks (ANN) to predict multivariate time series. The ANN model used in our operations contains three hidden layers, and each hidden layer consists of 50 hidden neurons.

• *VAR* [45]: This baseline uses the vector auto regression (VAR) to predict multivariate time series.

• *LSTM* [46]: This baseline uses the Long Short-Term Memory (LSTM) network to predict multivariate time series. The LSTM is a type of neural network specially designed for sequential data. This baseline treats the features and labels of the datasets as inputs and outputs of the LSTM model, respectively. The LSTM network used in our experiment contains one hidden layer that consists of 50 LSTM neurons.

• *ARIMA* [47]: We adopt the Auto Regressive Integrated Moving Average (ARIMA) model as a representative of univariate time series forecasting models. In the ARIMA experiments, we treat a multivariate time series as multiple univariate time series and use ARIMA models to predict their feature states, respectively.

• *LSTM-U*: This baseline uses LSTM as a univariate model to predict each series of a multivariate time series. This baseline is also a representative of univariate time series forecasting model. The LSTM network used in our experiment contains one hidden layer that consists of 50 LSTM neurons.

• *W-HFCM* [15]: The Wavelet-High-order FCM (W-HFCM) model is specially designed for univariate time series forecasting. It uses the redundant Haar wavelet transform to decompose original univariate time series as a multivariate time series and learns the parameters of HFCM by a method based on ridge regression. This baseline is a representative of the state-of-the-art univariate time series forecasting model based on the FCM framework.

• *Naive Method*: The naive method directly uses y_{t-n} as the predication of y_t. This is the simplest method in time series prediction. It is worth to note that the performance of the naive method is even acceptable for many scenarios.

• *Exponential Smoothing*: The exponential smoothing method calculates $s_t = \alpha y_t + (1-\alpha)s_{t-1}$ and uses s_t as a prediction of y_{t+1}, where α is the smoothing factor, and $0 < \alpha < 1$.

The hyper-parameters setting of the baselines are given in the Supplementary Materials.

3) Evaluation Metrics: We use two metrics to evaluate the model performance, including *Root Mean Square Error* (RMSE) and *Mean Absolute Error* (MAE), which are defined as follows:

$$\begin{aligned} \mathrm{RMSE} &= \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{x}^{(t)} - x^{(t)})^2}, \\ \mathrm{MAE} &= \frac{1}{T}\sum_{t=1}^{T}\left|\hat{x}^{(t)} - x^{(t)}\right|, \end{aligned} \tag{29}$$

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

TABLE III
COMPARISON WITH OTHER PREDICTION MODELS IN TERMS OF RMSE.

DataSet	Target Features	DFCM	FCM	ANN	VAR	LSTM	LSTM-U	ARIMA	W-HFCM	Naive	ES
Traffic	R1	**3.188**	7.958	3.736	4.152	3.401	3.402	4.081	3.320	4.188	14.66
	R2	**4.784**	10.64	5.109	6.000	4.820	5.335	6.458	5.206	6.561	19.87
	R3	4.359	7.603	4.902	4.981	4.388	4.642	5.120	**4.309**	5.387	13.09
	R4	**4.718**	10.87	5.360	7.060	4.986	5.317	7.342	4.911	7.346	23.52
	R5	**4.982**	11.25	5.147	7.073	5.270	5.609	7.204	5.101	7.202	22.65
	R6	**5.354**	11.25	5.961	7.945	5.776	6.219	8.317	5.582	8.354	25.38
AQIs	PM2.5	**12.73**	29.25	23.38	18.99	12.84	14.17	24.11	14.72	27.06	70.55
	NO2	**9.622**	16.01	12.06	14.85	9.657	9.716	19.35	9.900	22.24	42.66
	CO	**0.264**	0.728	0.719	0.346	0.269	0.270	0.440	0.617	0.487	1.012
	O3	**8.504**	15.30	13.02	14.47	8.713	9.995	16.93	20.86	20.72	40.11
	SO2	**2.732**	6.044	4.513	3.638	2.860	3.041	4.161	4.035	4.778	7.433
Power	Sub-metering_1	**2.141**	3.424	2.293	3.251	2.166	2.160	4.646	4.040	4.742	11.24
	Sub-metering_2	**2.752**	6.882	2.806	3.358	2.865	2.914	6.314	9.075	6.091	15.15
	Sub-metering_3	**1.069**	1.596	1.134	1.635	1.182	1.128	2.462	1.307	2.731	11.02
Temperature	Dining Room	**0.109**	1.821	0.488	0.121	0.140	0.139	0.195	0.208	0.316	2.492
	Room	**0.099**	1.709	0.580	0.130	0.143	0.207	0.207	0.197	0.324	2.516
Winning times		**15**	0	0	0	0	0	0	1	0	0
AVG arithmetic ranking		**1.063**	8.250	5.000	5.125	2.813	3.625	6.750	4.563	7.750	10.00
AVG geometric ranking		**1.044**	8.151	4.642	4.843	2.720	3.482	6.665	3.861	7.695	10.00

TABLE IV
COMPARISON WITH OTHER PREDICTION MODELS IN TERMS OF MAE.

DataSet	Target Features	DFCM	FCM	ANN	VAR	LSTM	LSTM-U	ARIMA	W-HFCM	Naive	ES
Traffic	R1	**2.467**	6.931	2.894	3.137	2.580	2.668	3.130	2.561	3.146	11.81
	R2	**3.593**	9.368	3.996	4.471	3.637	4.084	4.686	3.929	4.728	16.15
	R3	**3.171**	6.364	3.698	3.616	3.305	3.535	3.792	3.228	4.012	10.19
	R4	**3.114**	9.782	3.679	4.469	3.351	3.747	4.587	3.347	4.588	17.73
	R5	**3.405**	10.12	3.631	4.622	3.929	4.086	4.532	3.459	4.531	16.88
	R6	**3.245**	10.10	3.771	4.465	3.664	3.995	4.349	3.487	4.364	17.72
AQIs	PM2.5	**7.579**	22.03	16.26	10.92	8.646	10.07	13.52	9.950	15.48	51.75
	NO2	7.047	13.09	9.533	10.23	**6.715**	6.797	13.11	7.001	15.68	32.72
	CO	**0.161**	0.551	0.601	0.224	0.172	0.174	0.313	0.360	0.311	0.768
	O3	**6.349**	13.13	9.502	11.02	6.396	7.747	11.36	18.90	13.58	32.41
	SO2	**1.682**	3.619	3.063	2.335	1.863	2.381	2.566	2.687	2.758	5.017
Power	Sub-metering_1	**0.407**	1.625	0.733	0.550	0.567	0.500	0.958	0.663	0.692	4.274
	Sub-metering_2	**0.823**	2.516	0.915	0.848	0.885	1.198	1.477	2.857	1.393	6.902
	Sub-metering_3	**0.304**	0.994	0.445	0.370	0.561	0.456	0.543	0.447	0.595	7.209
Temperature	Dining Room	**0.085**	1.560	0.381	0.097	0.100	0.108	0.114	0.186	0.241	2.111
	Room	**0.076**	1.515	0.435	0.098	0.102	0.156	0.123	0.169	0.243	2.131
Winning times		**15**	0	0	0	1	0	0	0	0	0
AVG arithmetic ranking		**1.188**	8.625	5.625	4.625	2.938	4.063	6.313	4.438	7.188	10.00
AVG geometric ranking		**1.091**	8.594	5.273	4.123	2.697	3.901	6.220	3.827	7.112	10.00

where $x^{(t)}$ is the actual value of the state of a concept at time t, $\hat{x}^{(t)}$ is the predicted value, and T is the length of time series. The two metrics are the smaller, the better.

B. Results and Analysis

We present the results of all the comparison methods in Table III and Table IV with the summarized information listed in the bottom three lines. Note that the best performance for each dataset is emphasized in bold.

From the experiment results, we have three observations:

• The first is *the nonlinear models have superior performance over the linear models*. As shown in the tables, the methods with top three performance are DFCM, LSTM and LSTM-U. All of the three are deep neural-network-based forecasting models. Although Wavelet-HFCM adopts wavelet transform to extract frequency information from time series, its linear basic FCM framework can not fully exploit the information and therefore limits its performance.

• The second observation is *long-term dependencies is helpful for time series forecasting*. As shown in the tables, although the ANN model has nonlinear modeling abilities, its performance is still worse than Wavelet-HFCM. The reason might be that Wavelet-HFCM adopts historical data in a slice window as inputs to exploit long-term dependencies in time series, but the ANN model can only exploit dependencies between adjacent periods. The LSTM-based baselines can exploit long-term dependencies using memory gates [46], so their performance is better than ANN too. The basic FCM can neither exploit long-term dependencies nor model nonlinear

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

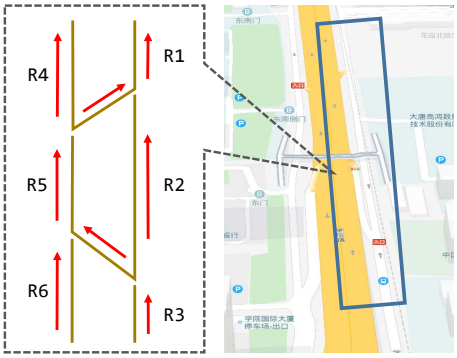


Fig. 3. A map of the road traffic system.

relationships, which leads its performance worse than all methods.

• The third observation is *the performance of multivariate models are better than univariate models*. Here, the performance of LSTM is better than LSTM-U, and the performance of VAR is better than ARIMA. The multivariate models treat multiple time series as a whole to exploit dependencies among them, so they have superior performance over univariate models that can only exploit dependencies inner a series.

In the DFCM model, neural network components enable it to model nonlinear relationships, the u-function adopts a LSTM network to exploit long-term dependencies, and the model can also exploit dependencies among different series. Therefore, as shown in the experiments, DFCM achieves the best performance in terms of both the largest number of wins (the best in 15 out of 16 prediction targets) and the smallest average performance ranks.

C. Effectiveness of Components

In this section, we perform detailed experiments to demonstrate the effectiveness of each component of the DFCM model. Due to space limit, we only report the results on the traffic and AQI datasets. The rest results show the similar findings, and are omitted here.

1) Baselines: In the experiments, we compare the prediction performances of five models:

- *DFCM-1L*: DFCM that contains one hidden layer in the f-function and one hidden layer in the u-function.
- *DFCM-3L*: DFCM that contains three hidden layers in the f-function and one hidden layer in the u-function.
- *f_i-1L*: DFCM that contains one hidden layer in the f-function and no u-function.
- *f_i-3L*: DFCM that contains three hidden layers in the f-function and no u-function.
- *FCM*: The basic FCM as a benchmark.

The comparison between DFCM-1L and DFCM-3L models is used to demonstrate the necessity of the deep neural network structure in the f-function. And, the comparison between DFCM-*L and f_i-*L are used to demonstrate the effectiveness of the u-function. Note that in the DFCM models, all hidden layers in both f and u consists of 40 neurons.

TABLE V
TRAFFIC SPEED PREDICTION IN TERMS OF RMSE.

	DFCM-1L	DFCM-3L	f_i-1L	f_i-3L	FCM
R1	3.249	**3.188**	6.044	5.034	7.958
R2	5.265	**4.784**	9.104	7.624	10.64
R3	4.485	**4.359**	6.958	5.958	7.603
R4	4.937	**4.718**	9.475	9.261	10.87
R5	5.188	**4.982**	9.132	8.674	11.25
R6	5.682	**5.354**	10.286	8.972	11.25
Overall	4.801	**4.564**	8.500	7.587	9.929

TABLE VI
TRAFFIC SPEED PREDICTION IN TERMS OF MAE.

	DFCM-1L	DFCM-3L	f_i-1L	f_i-3L	FCM
R1	2.531	**2.467**	4.417	3.760	6.931
R2	4.062	**3.593**	6.863	5.742	9.368
R3	3.372	**3.171**	4.978	4.420	6.364
R4	3.332	**3.114**	6.415	6.214	9.782
R5	3.699	**3.405**	6.191	5.988	10.12
R6	3.526	**3.245**	6.365	5.501	10.10
Overall	3.420	**3.166**	5.872	5.271	8.777

TABLE VII
CONCEPTS OF THE AIR POLLUTION SYSTEM.

Types	**Concepts**
AQIs	PM2.5, NO2, CO, O3, SO3
Meteorological	Temperature, Pressure, Humidity, Wind Speed

2) Prediction of Road Traffic System: The traffic dataset contains traffic speed data of six road segments of the Xueyuan Road in Beijing. Figure 3 illustrates the topological structure of the segments. In the experiments, the time line is divided evenly into slices, with ten minutes per slice. The period length τ is set to one day, *i.e.*, the natural rhythm of urban commuting. The results of DFCM and its variants are given in Table V and Table VI. From the results we have the following observations:

• First, all the variants of DFCM achieve much higher predictive accuracies than the basic FCM, which indeed performs very poorly. This implies that the basic FCM cannot well capture the complicated interactive relations between elements in a nonlinear real-world system. In contrast, by introducing non-linear neural networks into the f-function, DFCM works much better.

• Second, the performances of the DFCM models, *i.e.*, DFCM-1L and DFCM-3L, are obviously superior to that of the f_i benchmarks, *i.e.*, f_i-1L and f_i-3L. This indicates that the exogenous factors introduced to DFCM by the u-function is indeed of great help to the prediction of system states.

• Third, DFCM and f_i with three hidden layers in the f-function perform slightly better than that with only one layer. This implies that a deep structure is generally better than a shallow one, although excessive layers might lead to higher computational complexity and higher risk of overfitting.

In summary, the experiment results indicate that it is very effective to introduce deep neural networks into the FCM framework for improved prediction of nonlinear open systems.

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

TABLE VIII
AIR POLLUTION PREDICTION IN TERMS OF RMSE.

	DFCM-1L	DFCM-3L	f_i-1L	f_i-3L	FCM
PM2.5	18.95	**12.73**	21.42	20.425	29.25
NO2	10.46	**9.622**	10.68	10.39	16.01
CO	0.315	**0.264**	0.335	0.319	0.728
O3	9.346	**8.504**	14.61	12.95	15.30
SO2	4.246	**2.732**	5.008	4.609	6.044
Overall	6.143	**4.832**	7.361	6.864	10.47

TABLE IX
AIR POLLUTION PREDICTION IN TERMS OF MAE.

	DFCM-1L	DFCM-3L	f_i-1L	f_i-3L	FCM
PM2.5	12.72	**7.579**	16.59	14.40	22.03
NO2	7.536	**7.047**	7.671	7.606	13.09
CO	0.183	**0.161**	0.233	0.214	0.551
O3	7.125	**6.349**	11.41	9.837	13.13
SO2	2.212	**1.682**	3.446	3.221	3.619
Overall	4.278	**3.239**	5.569	4.986	8.360

3) Prediction of AQIs: Next, we run a similar experiment over the AQI dataset. As listed in Table VII, the dataset contains nine concepts, where the upper five are to describe air pollutants and the lower four to describe the meteorological factors. In the experiments, the period length τ of the u-function is also set as one day in the air pollution system. The prediction performance of DFCM and its variants are listed in Table VIII and IX, from which we have the following observations:

- First, the DFCM model with a three-layer f-function and a u-function (DFCM-3L) achieves the best prediction performance among all the competitors, which again verifies the superiority of DFCM in predictive applications.
- Second, the prediction performance of the models with three hidden layers (DFCM-3L and f_i-3L) is better than that with only one hidden layer (DFCM-1L and f_i-1L), which verifies the better predictive ability of deeper structures. Moreover, the performance of DFCM with u-functions (DFCM-1L and DFCM-3L) is better than that without u-functions (f_i-1L and f_i-3L), which again verifies the necessity of introducing exogenous factors for high-performance prediction. These results are consistent with that of the road traffic system.
- It is worth noting that the performance gap between DFCM with and without u-function is not as big as that in the traffic system case. This might be due to that the interrelationships between air pollutants and meteorological factors are more determinant than that between road segments in the traffic system.

In summary, the experimental results of the air pollution system indicate that the superior predictive power of DFCM is robust across different application scenarios and model parameter settings.

VI. INTERPRETATION EXPERIMENTS

In this section, we use the experiments over the AQI dataset to demonstrate the interpretation feature of DFCM. A similar experiments over the traffic dataset could be found in the supplementary materials.

TABLE X
THE AVERAGE INFLUENCE OF METEOROLOGICAL FACTORS TO AQIS IN THE BASIC FCM.

	PM2.5	NO2	CO	O3	SO2
Temp.	-0.66	-0.77	-0.79	0.32	-0.66
Pressure	-0.79	-0.73	-0.78	-0.67	-0.78
Humidity	-0.46	-0.41	-0.34	-0.30	-0.52
Wind	-0.53	-0.67	-0.54	-0.07	-0.46

TABLE XI
THE AVERAGE INFLUENCE ($w_{ij}^{(O)}$) OF METEOROLOGICAL FACTORS TO AIR POLLUTION.

	PM2.5	NO2	CO	O3	SO2
Temp.	-0.53	-0.85	-0.76	0.42	-0.36
Pressure	-0.39	-0.50	-0.53	-0.30	-0.14
Humidity	-0.37	-0.30	-0.28	-0.71	-0.29
Wind	-0.19	-0.55	-0.31	0.04	-0.21

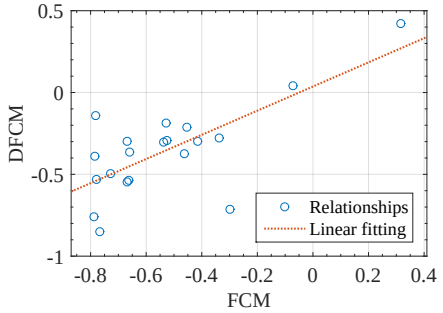


Fig. 4. The correlation of the relationships modeled by FCM and DFCM from the AQI dataset.

A. Interpretation of Relationships

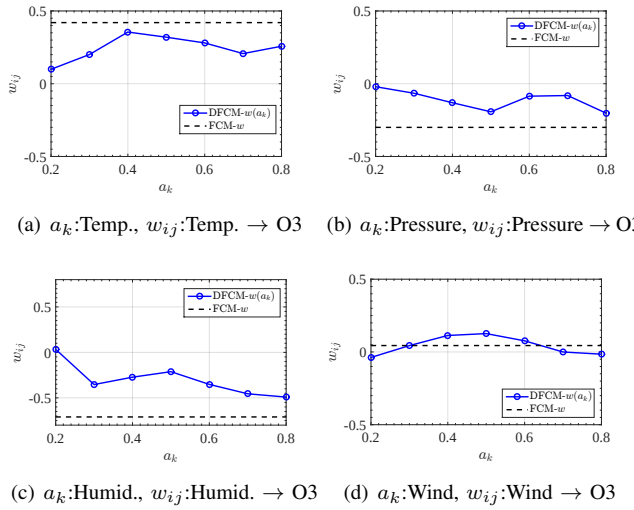
The concept of interpretation in DFCM is inherited from the basic FCM, where the relationships between concepts are interpreted as connection weights w_{ij} of FCM. Analogously, DFCM uses the Overall Relationship Strength $w_{ij}^{(O)}$ defined in Eq. (17) to interpret relationships among concepts. In this section, we use the AQI dataset as a show case to demonstrated how FCM and DFCM interpret the influences of meteorological factors to air pollution using w_{ij} and $w_{ij}^{(O)}$.

In the experiments, we use the AQI dataset to train FCM and DFCM models, respectively. The experiment setups are same as basic FCM and DFCM-3L in the section V-C3. After the model are well trained, we calculate $w_{ij}^{(O)}$ of DFCM according to Eq. (14) and extract w_{ij} from the basic FCM model. The $w_{ij}^{(O)}$ and w_{ij} are used to express (interpret) the influences from the concept i to the concept j.

Table X and Table XI list the strengths of influences measured by w_{ij} of FCM and by $w_{ij}^{(O)}$ of DFCM, respectively. Figure 4 plots the correlation of the values in the two tables. We can see that the values in the two tables are highly correlated. Indeed, the correlation between the two tables is 0.72, indicating the knowledge revealed by FCM and DFCM from the AQI dataset is very similar. Further, we can see for most of the conditions, the meteorological factors have

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems



(a) a_k:Temp., w_{ij}:Temp. → O3 (b) a_k:Pressure, w_{ij}:Pressure → O3

(c) a_k:Humid., w_{ij}:Humid. → O3 (d) a_k:Wind, w_{ij}:Wind → O3

Fig. 5. Influence of meteorological factors to air pollution.

negative impacts to the pollutants. This could be interpreted by some basic meteorological knowledge as follows:

- High air pressure in an area causes air flowing to surrounding areas and thus takes pollutants away. So air pressure has negative influence to air pollution.
- High humidity usually corresponds to precipitation weather, such as rain and snow, which can wash away pollutants in air. So humidity also has negative influence to air pollution.
- Wind can blow air pollutants away, so it is negative *w.r.t.* air pollution.
- The relationships between temperature and air pollutants are relatively indirect. In the northern cities of China, like Beijing, people burn fossil fuels in winter for heating, which therefore increases air pollutants in low temperature days. Indeed as reported in [48], heavy pollution days usually appear in winter.

Nevertheless, not all relationships between the meteorological factors and pollutants follow the above rules. We can see that O3 has several special characteristics in the tables. In order to further study the influences of the meteorological factors to O3, we plots the conditional relationship strength, *i.e.*, $w_{ij}^{(C)}(a_k)$ defined in Eq. (17), for different meteorological factors to O3 in Fig. 5. The horizontal axis a_k denotes value of meteorological factors. The vertical axis denotes influences $w_{ij}^{(C)}$ of the meteorological factors with corresponding meteorological factor values as conditions.

As shown in Fig. 5(b) and 5(c), both air pressure and humidity have negative influence to O3 for different values of a_k, which is consistent with that for other pollutants. Fig. 5(a) however shows that the temperature has positive influence to O3, which could be due to the fact that high temperatures can facilitate the production of O3 [49]. The relationship between wind speed and O3 is more complicated. When the wind speed is not very high, the airflow of wind can blow O3 pollutants away and then reduces the concentration of O3. When the wind speed is high, it can reduce the stability of the boundary layer of atmosphere, and the intrusion of O3 from the upper layer to the surface layer can thus increase the O3 concentration of

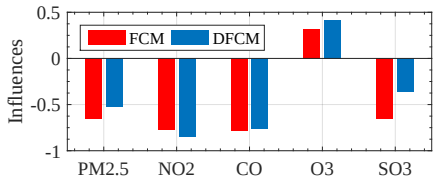
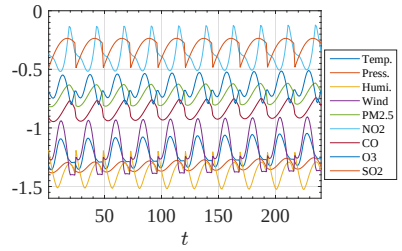
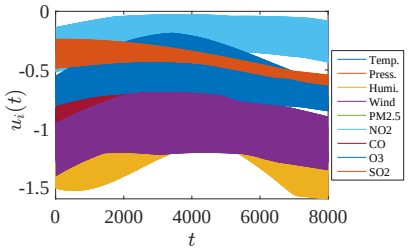


Fig. 6. Expressing the influence of temperature to AQIs using a bar map.



(a) Short-Term Periodicity



(b) Long-Term Periodicity

Fig. 7. Periodicity of $u_i(t)$ in the air pollution system.

the surface [49]. Therefore, as shown in Fig. 5(d), when the value of a_k is very low, the influence $w_{ij}^{(C)}$ of wind speed to O3 is in negative, but when the value of a_k becomes higher, $w_{ij}^{(C)}$ turns into positive values. More extremely, when the wind speed is too high, the high dispersion of wind could offset the O3 contributions from the upper layer. Therefore, the influence $w_{ij}^{(C)}$ reduces to a value near to zero when a_k is very high. It is interesting that the results about the relationships between wind speed and O3 agree with the conclusion of related meteorology studies [49].

Comparing the results in Table X and Fig. 5, we can see the relationships modeled by DFCM contain more detail information, then could be used in nonlinear system analysis. Oppositely, the basic FCM model can only express linear relationships, and is not very suitable to complex system analysis. For simple concept relationships, a more comprehensible way to express the influences is using the bar map. Figure 6 shows the influences of temperature to AQIs. The temperature has significant negative influences to PM2.5, NO2, CO, and SO3, while has positive to O3.

B. Interpretation of Exogenous Factors

DFCM uses the u-function to model exogenous factors of a system. Therefore, we can use the value of u-function to interpret influences of exogenous factors to system concepts. Fig. 7 depicts the periodicity of $u_i(t)$ for different meteorological factors and pollutants. In Fig. 7(a), we plot the fluctuations

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

of $u_i(t)$ in the first ten days. As can be seen, there is an obvious rhythm in $u_i(t)$ that repeats every day. Fig. 7(b) plots the fluctuations of $u_i(t)$ in one year. As shown in the figure, although we do not adopt any seasonal information other than τ in the u-function, there exists long-term seasonal periodicity that waves from spring to winter for each concept. Accordingly, we might expect that the short-term periodicity in $u_i(t)$ corresponds to the influence of human activities to air pollution since life rhythms of human are mostly with daily frequency. The long-term periodicity might correspond to the impact of season changes to air pollution. In the Deep FCM model, the influences of different exogenous factors are all integrated into a same u-function.

The results in the performance and interpretation experiments demonstrate that the proposed DFCM model not only have high predictive ability for multivariate time series forecast tasks but also can offer meaningful explanations for the prediction results.

VII. Conclusion and Future Works

In this paper, we proposed a deep neural network-based fuzzy cognitive maps model, named Deep FCM, to gain interpretable multivariate prediction. The Deep FCM model introduces deep neural network models into the knowledge representation framework of FCM so that the advantage of FCM in interpretation and the advantage of deep neural networks in prediction can be integrated into a same model. The excellent performances of Deep FCM in terms of both interpretability and predictive ability were verified over two real-world open systems. Deep FCM indeed provided an important clue to building interpretable predictors for real-life applications.

How to handle exogenous factors is an age-old challenge for system modelling and attribution analysis studies [38]. Our DFCM model is designed for time series prediction tasks, so we pay special attention to time-related factors and introduce a LSTM-based u-function to capture the exogenous factors with time dependence. Nevertheless, DFCM is still an open framework that calls for new approaches for modelling other types of exogenous factors. Studying new exogenous factors modeling approaches is our next step work.

Acknowledgments

This work was supported by the National Key R&D Program of China (2019YFB2102100). Dr. Jingyuan Wang's work was partially supported by the National Natural Science Foundation of China (Grant No.61572059), the Fundamental Research Funds for the Central Universities (YWF-20-BJ-J-839). Prof. Junjie Wu's work was partially supported by National Natural Science Foundation of China (71531001,71725002, U1636210). Prof. Zhen Peng's work was partially supported by National Science Foundation of China (No.71601022), the Youth Top Talent Cultivation Plan Project of Beijing (No.CIT&TCD201804036).

Appendix A
Derivations of AFGD Algorithm

The objective function of the AFGD algorithm is

$$\min L(\theta_f, \theta_u) = \frac{1}{2}\sum_{t=1}^{T}\left(f_j(\boldsymbol{a}^{(t)}|\theta_f) + u_j(t|\theta_u) - z_j^{(t)}\right)^2. \quad (30)$$

In the AFDG algorithm, we optimize the parameters θ_f and θ_u iteratively. Given a training sample $\{\boldsymbol{a}^{(t)}, z_j^{(t)}\}$, we denote $f_j^{(t)}(\theta_f) = f_j(\boldsymbol{a}^{(t)}|\theta_f)$ and $u_j^{(t)}(\theta_u) = u_j(t|\theta_u)$ for short. In the q-th round of iteration, the algorithm updates θ_f and θ_u to make following equations hold:

$$\begin{aligned} f_j^{(t)}(\theta_f^{(q)}) &= f_j^{(t)}(\theta_f^{(q-1)}) - \eta_f \cdot \left.\frac{\partial L(f_j)}{\partial f_j}\right|_{f_j = f_j^{(t)}(\theta_f^{(q-1)})}, \\ u_j^{(t)}(\theta_u^{(q)}) &= u_j^{(t)}(\theta_u^{(q-1)}) - \eta_u \cdot \left.\frac{\partial L(u_j)}{\partial u_j}\right|_{f_j = f_j^{(t)}(\theta_u^{(q-1)})}, \end{aligned} \quad (31)$$

where $\theta_f^{(q)}$ and $\theta_u^{(q)}$ are the parameters in the q-th iteration, and η_f, η_u are learning rates. Eq. (31) ensures that the parameters updating direction is along the negative gradient direction of the loss function to the functions f_j and u_j, so we name our algorithm as *Function Gradient Descent*.

According to the definition of the loss function in Eq. (30), for any training data $z_j^{(t)}$, the partial derivatives of $L(f_j, u_j)$ to f_j and u_j are in the form of

$$\begin{aligned} \left.\frac{\partial L(f_j)}{\partial f_j}\right|_{f_j = f_j^{(t)}(\theta_f^{(q-1)})} &= z_j^{(t)} - u_j^{(t)}(\theta_u^{(q-1)}) - f_j^{(t)}(\theta_f^{(q-1)}), \\ \left.\frac{\partial L(u_j)}{\partial u_j}\right|_{f_j = f_j^{(t)}(\theta_u^{(q-1)})} &= z_j^{(t)} - f_j^{(t)}(\theta_f^{(q-1)}) - u_j^{(t)}(\theta_u^{(q-1)}). \end{aligned} \quad (32)$$

Plugging Eq. (32) into Eq. (31), we have

$$\begin{aligned} f_j^{(t)}(\theta_f^{(q)}) =& f_j^{(t)}(\theta_f^{(q-1)}) \\ &+ \eta_f\left(z_j^{(t)} - u_j^{(t)}(\theta_u^{(q-1)}) - f_j^{(t)}(\theta_f^{(q-1)})\right), \end{aligned} \quad (33)$$

$$\begin{aligned} u_j^{(t)}(\theta_u^{(q)}) =& u_j^{(t)}(\theta_u^{(q-1)}) \\ &+ \eta_u\left(z_j^{(t)} - f_j^{(t)}(\theta_f^{(q-1)}) - u_j^{(t)}(\theta_u^{(q-1)})\right). \end{aligned} \quad (34)$$

For the f-function, we plug Eq. (33) into Eq. (30), the loss function L in the q-th iteration is in the form of

$$\begin{aligned} L(\eta_f) = \frac{1}{2}\sum_{t=1}^{T}\Big[& f_j^{(t)}(\theta_f^{(q-1)}) + u_j^{(t)}(\theta_u^{(q-1)}) - z_j^{(t)} \\ &+\eta_f\left(z_j^{(t)} - u_j^{(t)}(\theta_u^{(q-1)}) - f_j^{(t)}(\theta_f^{(q-1)})\right)\Big]^2 \end{aligned} \quad (35)$$

For the learning rate η_f, the function in Eq. (35) is a convex function, so the optimum point of L can be achieved at $\partial L/\partial \eta_f = 0$, *i.e.,*

$$\begin{aligned} \frac{\partial L}{\partial \eta_f} = \sum_{t=1}^{T}\Bigg[&\Big(f_j^{(t)}(\theta_f^{(q-1)}) + u_j^{(t)}(\theta_u^{(q-1)}) - z_j^{(t)} + \\ &\eta_f\left(z_j^{(t)} - u_j^{(t)}(\theta_u^{(q-1)}) - f_j^{(t)}(\theta_f^{(q-1)})\right)\Big) \\ &\times \underbrace{\left(z_j^{(t)} - u_j^{(t)}(\theta_u^{(q-1)}) - f_j^{(t)}(\theta_f^{(q-1)})\right)}_{\text{TermI}}\Bigg] = 0. \end{aligned} \quad (36)$$

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

We cannot have $\left(z_j^{(t)} - u_j^{(t)}(\theta_u^{(q-1)}) - f_j^{(t)}(\theta_f^{(q-1)})\right) = 0$ held for all samples for Term I, so the following equation need be satisfied:

$$f_j^{(t)}(\theta_f^{(q-1)}) + u_j^{(t)}(\theta_u^{(q-1)}) - z_j^{(t)} + \eta_f \left(z_j^{(t)} - u_j^{(t)}(\theta_u^{(q-1)}) - f_j^{(t)}(\theta_f^{(q-1)})\right) = 0 \tag{37}$$

which means $\eta_f = 1$. By plugging $\eta_f = 1$ into Eq. (33), we have

$$f_j^{(t)}(\theta_f^{(q)}) = z_j^{(t)} - u_j^{(t)}(\theta_u^{(q-1)}). \tag{38}$$

For the u-function, we also have $\eta_u = 1$ via the similar derivations as that in Eqs. (33) - (37). By plugging $\eta_u = 1$ into Eq. (34), we have

$$u_j^{(t)}(\theta_u^{(q)}) = z_j^{(t)} - f_j^{(t)}(\theta_f^{(q-1)}). \tag{39}$$

Therefore, Eq. (31) (*i.e.,* Eq. (26)) is equal to following parameter iteration functions:

$$f_j(\boldsymbol{a}^{(t)}|\theta_f^{(q)}) = z_j^{(t)} - u_j(t|\theta_u^{(q-1)}), \tag{40}$$

$$u_j(t|\theta_u^{(q)}) = z_j^{(t)} - f_j(\boldsymbol{a}^{(t)}|\theta_f^{(q)}). \tag{41}$$

REFERENCES

[1] B. Kosko, "Fuzzy cognitive maps," *International journal of man-machine studies*, vol. 24, no. 1, pp. 65–75, 1986.

[2] K. Perusich, "Fuzzy cognitive maps for policy analysis," in *Proceedings of 1996 International Symposium on Technology and Society Technical Expertise and Public Decisions*. IEEE, 1996, pp. 369–373.

[3] Z. Wei, L. Lu, and Z. Yanchun, "Using fuzzy cognitive time maps for modeling and evaluating trust dynamics in the virtual enterprises," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1583–1592, 2008.

[4] V. C. Georgopoulos, G. A. Malandraki, and C. D. Stylios, "A fuzzy cognitive map approach to differential diagnosis of specific language impairment," *Artificial intelligence in Medicine*, vol. 29, no. 3, pp. 261–278, 2003.

[5] A. S. Andreou, N. H. Mateou, and G. A. Zombanakis, "The cyprus puzzle and the greek-turkish arms race: Forecasting developments using genetically evolved fuzzy cognitive maps," *Defence and Peace Economics*, vol. 14, no. 4, pp. 293–310, 2003.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[7] Y. Miao, Z.-Q. Liu, C. K. Siew, and C. Y. Miao, "Dynamical cognitive network-an extension of fuzzy cognitive map," *IEEE transactions on Fuzzy Systems*, vol. 9, no. 5, pp. 760–770, 2001.

[8] K. S. Park and S. H. Kim, "Fuzzy cognitive maps considering time relationships," *International Journal of Human-Computer Studies*, vol. 42, no. 2, pp. 157–168, 1995.

[9] Y. Cai, C. Miao, A.-H. Tan, Z. Shen, and B. Li, "Creating an immersive game world with evolutionary fuzzy cognitive maps," *IEEE computer graphics and applications*, vol. 30, no. 2, pp. 58–70, 2010.

[10] J. L. Salmeron, "Modelling grey uncertainty with fuzzy grey cognitive maps," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7581–7588, 2010.

[11] D. K. Iakovidis and E. Papageorgiou, "Intuitionistic fuzzy cognitive maps for medical decision making," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 1, pp. 100–107, 2011.

[12] Z. Chunying, L. Lu, O. Dong, and L. Ruitao, "Research of rough cognitive map model," in *Advanced Research on Electronic Commerce, Web Application, and Communication*. Springer, 2011, pp. 224–229.

[13] J. P. Carvalho and J. A. B. Tome, "Rule based fuzzy cognitive maps-expressing time in qualitative system dynamics," in *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 280–283.

[14] M. Hagiwara, "Extended fuzzy cognitive maps," in *Fuzzy Systems, 1992., IEEE International Conference on*. IEEE, 1992, pp. 795–801.

[15] S. Yang and J. Liu, "Time-series forecasting based on high-order fuzzy cognitive maps and wavelet transform," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3391–3402, 2018.

[16] E. I. Papageorgiou and K. Poczeta, "A two-stage model for time series prediction based on fuzzy cognitive maps and neural networks," *Neurocomputing*, vol. 232, pp. 113–121, 2017.

[17] S. Hengjie, M. Chunyan, W. Roel, and F. Catthoor, "Implementation of fuzzy cognitive maps based on fuzzy neural networks and application in numerical prediction of time series," *IEEE Trans. Fuzzy Systems*, vol. 18, pp. 233–250, 2010.

[18] H. Song, C. Miao, Z. Shen, W. Roel, D. Maja, and C. Francky, "Design of fuzzy cognitive maps using neural networks for predicting chaotic time series," *Neural Networks*, vol. 23, no. 10, pp. 1264–1275, 2010.

[19] W. Pedrycz, "Why triangular membership functions?" *Fuzzy sets and Systems*, vol. 64, no. 1, pp. 21–30, 1994.

[20] E. Papageorgiou, C. Stylios, and P. Groumpos, "Fuzzy cognitive map learning based on nonlinear hebbian rule," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2003, pp. 256–268.

[21] E. Papageorgiou, C. D. Stylios, and P. P. Groumpos, "Active hebbian learning algorithm to train fuzzy cognitive maps," *International journal of approximate reasoning*, vol. 37, no. 3, pp. 219–249, 2004.

[22] D. Koulouriotis, I. Diakoulakis, and D. Emiris, "Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behavior," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 1. IEEE, 2001, pp. 364–371.

[23] K. E. Parsopoulos, E. I. Papageorgiou, P. Groumpos, M. N. Vrahatis *et al.*, "A first study of fuzzy cognitive maps learning using particle swarm optimization," *Proc. IEEE*, pp. 1440–1447, 2003.

[24] E. I. Papageorgiou and J. L. Salmeron, "A review of fuzzy cognitive maps research during the last decade," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 66–79, 2013.

[25] H. J. Song, C. Y. Miao, R. Wuyts, Z. Q. Shen, M. D Hondt, and F. Catthoor, "An extension to fuzzy cognitive maps for classification and prediction," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 116–135, 2011.

[26] X. Lai, Y. Zhou, and W. Zhang, "Software usability improvement: modeling, training and relativity analysis," in *Information Science and Engineering (ISISE), 2009 Second International Symposium on*. IEEE, 2009, pp. 472–475.

[27] D. Ruan, F. Hardeman, and L. Mkrtchyan, "Using belief degree-distributed fuzzy cognitive maps in nuclear safety culture assessment," in *Fuzzy Information Processing Society (NAFIPS), 2011 Annual Meeting of the North American*. IEEE, 2011, pp. 1–6.

[28] Y. Miao, C. Miao, X. Tao, Z. Shen, and Z. Liu, "Transformation of cognitive maps," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 114–124, 2010.

[29] J. Aguilar, "A dynamic fuzzy-cognitive-map approach based on random neural networks," *International Journal of Computational Cognition*, vol. 1, no. 4, pp. 91–107, 2003.

[30] G. Acampora, V. Loia, and A. Vitiello, "Distributing emotional services in ambient intelligence through cognitive agents," *Service Oriented Computing and Applications*, vol. 5, no. 1, pp. 17–35, 2011.

[31] W. Stach, L. A. Kurgan, and W. Pedrycz, "Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 61–72, 2008.

[32] W. Homenda, A. Jastrzebska, and W. Pedrycz, "Modeling time series with fuzzy cognitive maps," in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2014, pp. 2055–2062.

[33] W. Pedrycz, A. Jastrzebska, and W. Homenda, "Design of fuzzy cognitive maps for modeling time series," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 1, pp. 120–130, 2016.

[34] F. Vanhoenshoven, G. Nápoles, S. Bielen, and K. Vanhoof, "Fuzzy cognitive maps employing arima components for time series forecasting," in *International Conference on Intelligent Decision Technologies*. Springer, 2017, pp. 255–264.

[35] E. I. Papageorgiou and W. Froelich, "Multi-step prediction of pulmonary infection with the use of evolutionary fuzzy cognitive maps," *Neurocomputing*, vol. 92, pp. 28–35, 2012.

[36] W. Froelich and E. I. Papageorgiou, "Extended evolutionary learning of fuzzy cognitive maps for the prediction of multivariate time-series," in *Fuzzy cognitive maps for applied sciences and engineering*. Springer, 2014, pp. 121–131.

[37] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[38] A. W. Kruglanski, "The endogenous-exogenous partition in attribution theory." *Psychological Review*, vol. 82, no. 6, p. 387, 1975.

[39] B. F. Hobbs, S. A. Ludsin, R. L. Knight, P. A. Ryan, J. Biberhofer, and J. J. Ciborowski, "Fuzzy cognitive mapping as a tool to define management objectives for complex ecosystems," *Ecological Applications*, vol. 12, no. 5, pp. 1548–1565, 2002.

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2020.3005293, IEEE Transactions on Fuzzy Systems

[40] O. Osoba and B. Kosko, "Beyond dags: Modeling causal feedback with fuzzy cognitive maps," *arXiv preprint arXiv:1906.11247*, 2019.
[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[42] D. E. Rumelhart, G. E. Hinton, R. J. Williams *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.
[43] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat, "Genetic learning of fuzzy cognitive maps," *Fuzzy sets and systems*, vol. 153, no. 3, pp. 371–401, 2005.
[44] A. L. S. Maia, F. d. A. de Carvalho, and T. B. Ludermir, "Forecasting models for interval-valued time series," *Neurocomputing*, vol. 71, no. 16-18, pp. 3344–3352, 2008.
[45] T. Abeysinghe, U. Balasooriya, and A. Tsui, "Small-sample forecasting regression or arima models?" *Journal of Quantitative Economics*, vol. 1, no. 1, pp. 103–113, 2003.
[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[47] R. H. Shumway and D. S. Stoffer, *Time series analysis and its applications: with R examples*. Springer, 2017.
[48] J. Quan, X. Tie, Q. Zhang, Q. Liu, X. Li, Y. Gao, and D. Zhao, "Characteristics of heavy aerosol pollution during the 2012–2013 winter in beijing, china," *Atmospheric Environment*, vol. 88, pp. 83–89, 2014.
[49] J. Tu, Z.-G. Xia, H. Wang, and W. Li, "Temporal variations in surface ozone and its precursors and meteorological effects at an urban site in china," *Atmospheric Research*, vol. 85, no. 3-4, pp. 310–337, 2007.

Jingyuan Wang received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He is currently an Associate Professor of School of Computer Science and Engineering, Beihang University, Beijing, China. His is also the head of Beihang Interest Group on SmartCity (BIGSCity), and Vice Director of the Beijing City Lab (BCL). His general area of research is data mining and machine learning, with special interests in smart cities, finance and healthcare data analytics.

Zhen Peng received the B.E. and M. E. degree in computer science and technology from Shandong University and Ph.D degree in computer application technology from University of Science and Technology Beijing. She is currently a Professor in Information Management and Information System Department, Beijing Institute of Petrochemical Technology, the director of Information Management and Information System Department, and the vice director of Energy Economic Center. Her general area of research is data mining and fuzzy cognitive maps, with special interests in air pollution analytics.

Xiaoda Wang received the B.E. degree in Beijing Institute of Technology, Beijing, China in 2017. Since then, he has been working toward the M.E. degree in School of Computer Science and Engineering, Beihang University. His current research interests include AI, data mining, and urban data analysis.

Chao Li received his Ph.D degree from Beihang University. He is an associate professor at School of Computer Science and Engineering, Beihang University, China. His main research interests include the Smart City, data vitalization and multimedia applications.

Junjie Wu is currently a full professor in Information Systems Department, School of Economics and Management, Beihang University. He is also the director of the Research Center for Data Intelligence (DIG), the vice director of Beijing Key Laboratory of Emergency Support Simulation Technologies for City Operations, and a senior researcher in Beijing Innovation Center for Big Data and Brain Computing. His general area of research is data mining and machine learning, with a special interest in social, urban and financial computing. He is the recipient of various nation-wide academic awards in China, including the NSFC Distinguished Young Scholars, the MOE Changjiang Young Scholars, and the National Excellent Doctoral Dissertation.

1063-6706 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 14:21:20 UTC from IEEE Xplore. Restrictions apply.