1 INTRODUCTION

MULTI-STEP time series prediction is a major topic in data analytics and forecasting and brings significant impact to multiple fields [1], such as finance [2], transportation [3], meteorology [4], [5], and power industry [6]. In these fields, making critical operational decisions often depends on obtaining accurate predictions for multiple future data points. In multivariate time series, exogenous variables make different contributions to the target series [7], [8], [9]. In the context of deep neural networks, such as recurrent neural networks (RNNs) [10], [11], [12] which are popular time series predictors, variable contributions can be weakened by the network’s processing mode which blends the information from exogenous variables into non-transparent hidden states. Other challenges in continuous prediction include the provision of interpretability and prediction insight, as well as building continuous networks for predicting multiple future values at arbitrary time points.

Currently, the basic idea behind interpretability is to develop an attention mechanism to distinguish the different contributions among exogenous variables [13], [14]. Although existing explainable methods achieve good prediction performance, they have certain limitations for multi-step predictions. One issue is that discretizing observations and emission intervals is required for building discrete neural networks. Such networks can only make predictions for multiple future values at discrete steps and the predictions have the same gap as the input data. In contrast to discrete networks, continuous ones circumvent this limitation and can predict an entire time interval depending on the forecasting demands of the particular application, even if such time resolution is not available in the recorded training dataset where the latter only contains samples at regular or discrete time instances. Fig. 1 exemplifies the different predictive styles of discrete and continuous neural networks. An example for applying continuous networks in the real world is the distribution of electricity consumption which is relatively higher during daytime on weekends. Through continuous electricity consumption prediction, electric power companies could allocate resources more precisely in electricity generation to maximize resource utilization, reduce costs and promote smart grid development.

In this work, we introduce a novel explainable continuous neural network for arbitrary-step prediction as overviewed in Fig. 2. Its framework involves ordinary differential equations (ODE) to achieve continuous-time prediction by
A novel ETN-ODE framework is proposed which, to
We design a tandem attention mechanism to gener-
W
. In addition to predicting, the provi-
ELATED
. Although traditional methods, such as
We develop a TGRU to represent the different
2
þ
þ
þ
0
3
1
¼
1
with arbitrary steps
þ
26x482
M
26x491
Continuous neural networks make predictions at continuous time points
contributions of exogenous variables are captured to
variable attention, different
attention is employed to capture the contributions in the
time domain. With respect to variable attention, different
influence on the target series with short historical data while
others with long historical data, and thus the temporal
attention is employed to capture the contributions in the
time domain. With respect to variable attention, different
capacity into the input data and control network information
furthermore, the tandem attention with respect to temporal
exogenous variables represents one of our contributions.
Furthermore, the tandem attention with respect to temporal
and variable attention is designed to provide interpretabil-
ity into the input data and control network information
flow. Some exogenous variables might have significant
influence on the target series with short historical data while
others with long historical data, and thus the temporal
attention is employed to capture the contributions in the
time domain. With respect to variable attention, different
capabilities of exogenous variables are captured to
parameterizing the derivatives of the latent states. The raw
time series is encoded by a specially designed tensorized
gated recurrent unit (TGRU). Additionally, a tandem atten-
tion mechanism is incorporated to enhance the encoding ability and provide more adaptive input representations to the
ODE network, and also capture the different contributions among the exogenous variables. In multi-step prediction,
modeling the incremental relationships can help forecast future values at arbitrary time points with continuous neural
networks. As ODEs can capture the relationships between unknown functions and their derivatives, such modeling
allows neural networks to support continuous operation and obtain future values at arbitrary forecasting steps. In many
real-world applications, building continuous networks is sufficient to reduce the costs when high-frequency predictions
are needed with low-frequency sampled input data.

To generate a better mapping of the hidden states for the proposed network, we design a TGRU and a tandem atten-
tion mechanism to process the multivariate inputs. First, the
proposed TGRU modifies the internal structure of the basic
GRU cell which processes the multivariate input together, and combines them into a single structure. To capture dif-
ferent dynamics of individual input features, we design the
hidden states as a matrix in a single cell. Each series is pro-
jected to an independent hidden space. Consequently, our
model can be adaptable to the multivariate input and capa-
ble of enhancing the contribution of each exogenous indi-
gual series. Improved modeling of the impact of exogenous variables represents one of our contributions.
Furthermore, the tandem attention with respect to temporal
and variable attention is designed to provide interpretability into the input data and control network information
flow. Some exogenous variables might have significant
influence on the target series with short historical data while
others with long historical data, and thus the temporal
attention is employed to capture the contributions in the
time domain. With respect to variable attention, different
capabilities of exogenous variables are captured to

represent different predictive abilities. Finally, the tandem
attention and the hidden states are formed to produce a con-
text vector being the input of a continuous-time ODE net-
work making at arbitrary time point predictions. Ablation
studies show the effectiveness of each component of the
proposed ETN-ODE architecture in the experimental
section.

Our major contributions are summarized in the following:

- A novel ETN-ODE framework is proposed which, to the
  best of our knowledge, is the first explainable continuous
  neural network for multi-step time series prediction at
  arbitrary time instances for multiple variables. In our
  experiments, it shows to be substantially more accurate than
  current ODE models for arbitrary-step prediction.

- We design a tandem attention mechanism to generate
  a more adaptive input to the ODE network, which offers additional interpretability to the temporal
  and variable contribution to the target series when forecasting at both discrete and continuous
  steps.

- We develop a TGRU to represent the different
dynamics of the individual series. Compared with
the commonly used tensorized long-short term memory
(LSTMs), the employed TGRU here allows
fewer model parameters and this further improves
prediction performance.

2 RELATED WORK

Time Series Prediction. Although traditional methods, such as
the Autoregressive (AR) model [15] and the Vector Autore-
gressive model [16], have shown effectiveness in many real-
world applications they cannot model nonlinear relation-
ships for multivariate time series. In the last decade, deep
learning methods for time series prediction have been
developed significantly due to their ability for dealing with
nonlinearities [17], [18]. Another advantage of such meth-
ods for real-world applications is their ability to predict
multiple values given historical information [19], [20], [21].
An example is the work in [22] that used an LSTM based
architecture to forecast multiple values on web traffic data.

Attention Mechanism. In addition to predicting, the provi-
sion of interpretability is also important when processing
multivariate time series. Examples include two types of
attention mechanisms generated with independent RNNs
in [14], [23]. A mixture attention mechanism was also

Fig. 2. Overview of the proposed ETN-ODE structure for arbitrary-step
time series prediction.

Fig. 1. (a) Discrete neural networks make predictions at time points T + 1,
T + 2, and T + 3, which have the same sample gap as the input data. (b) Continuous neural networks make predictions at continuous time points T +
M with arbitrary steps M = 0.3, 1.7, etc.
proposed in [13] aiming at enhancing interpretability with a tensorized LSTM structure for single-step predictions. Recently, the idea of tensorizing hidden states has shown to be advantageous in multivariate predictions [24], [25], as the attention mechanism can enable to distinguish the different contributions among exogenous variables.

Neural Networks with ODEs. Recently, the neural ODE work in [26] proposed a continuous model for the prediction of values at arbitrary time points in univariate time series. Such a model was applied to irregularly sampled time series classification in [27], where an ODE-RNN encoder was designed to process the irregularly sampled data instead of fixing the sample gap. The work in [28] presented a neural ODE control (NODEC) framework for learning feedback control signals on graphs. Another example is [29] for the processing of sporadically observed time series, which combined Bayes method and ODE networks for prediction, missing value imputation and reconstruction. In general, none of the existing continuous models deals with multivariate time series for multi-step prediction as it is the case for the proposed ETN-ODE model here, which also incorporates an attention mechanism for interpretability in multi-step predictions for multivariate time series.

3 MAIN METHOD

This section presents the introduced ETN-ODE. The model structure consists of a TGRU, a tandem attention mechanism, and an ODE network. As shown in Fig. 3, the TGRU is used to process the multivariate time series. Two types of attention mechanisms are used to process the hidden states and generate the context vector $C_T$. This vector is used to produce an initial latent value $z_T$ assumed to have a Gaussian distribution $\mathcal{N}(0, I)$. Given the initial value and a prediction time interval, the ODE network generates multiple future values as shown in Fig. 4.

3.1 Problem Statement

Most of the deep learning models for time series prediction forecast at time $T+1$ given historical samples from $T$ previous steps. Our aim here is to forecast arbitrary multiple future values with continuous time intervals. Making predictions at arbitrary points can also be employed for fixing historical data with missing values or deal with irregularly sampled data to forecast high-frequency future values with less data. Assume that we have a target series $Y$ of length $T$ where $Y = [y_1, y_2, \ldots, y_T]^T \in \mathbb{R}^T$. Given the prediction time interval $\{T+m_1, \ldots, T+m_K\}$, our aim is to design a non-linear mapping to forecast multiple future values of the target series represented as $[\hat{y}_{T+m_1}, \ldots, \hat{y}_{T+m_K}]^T = \mathcal{F}(X_T)$, where $K$ is the number of predicted values. The predicted time interval can be set to continuous values during the testing stage for arbitrary-step prediction, e.g., $\{T + 0.3, \ldots, T + 2.6\}$. Here $\mathcal{F}(\cdot)$ is achieved by an explainable continuous framework and $X_T = \{x_1, x_2, \ldots, x_T\}$ is the multivariate input data, where $x_t = [x_{t}^1, x_{t}^2, \ldots, x_{t}^N, y_t]^T$, for $t = 1, \ldots, T$. Thus, we have $N + 1$ model input features,
where $X_T \in \mathbb{R}^{T \times (N+1)}$. The target series is related to $N$ exogenous series. Considering the autoregressive effectiveness of the target series, to form the multivariate input features we integrate $y_t$ and the $N$ exogenous variables at each time step of length $T$.

### 3.2 Tensorized GRU

For multivariate time series, the internal structure of the GRU cell may need certain specialization to obtain a better mapping in hidden space. Here, we tensorize the hidden state in order to learn the independent representation of individual series based on their specific information content, which is adaptable for the multivariate input and enhances the contribution of each individual series. The hidden state of the proposed TGRU is defined as a matrix rather than a vector, which is different from the typical GRU setup and involves more parameters to represent the hidden states. However, the TGRU is efficient as it only has an update and a reset gate compared to the tensorized LSTM which leads to an average of 20% fewer parameters to be learned. Results in the ablation study demonstrate the effectiveness of TGRU. The TGRU layer can be considered as a set of parallel GRUs with each of them processing one individual series.

To represent the tensorized GRU cell, we define the hidden state at time step $t$ as the matrix $H_t = [h^1_t, \ldots, h^{N+1}_t] \in \mathbb{R}^{(N+1) \times d}$, where $h^n_t \in \mathbb{R}^d$ is the hidden state vector mapping the $n$th individual input feature at time step $t$ for $n = 1, \ldots, N + 1$, and $d$ is the hidden dimension of each input feature.

According to standard GRU networks, given incoming input data $x_t$ and the previous hidden state matrix $H_{t-1}$, the cell update consists of:

$$R_t = \sigma(W_r \circ H_{t-1} + V_r \circ x_t + b_r),$$  

$$U_t = \sigma(W_u \circ H_{t-1} + V_u \circ x_t + b_u),$$  

$$\tilde{H}_t = \tanh(W_h \circ (R_t \circ H_{t-1}) + V_h \circ x_t + b_h),$$  

$$H_t = (1 - U_t) \circ H_{t-1} + U_t \circ \tilde{H}_t,$$

where $R_t$ is the reset gate, $U_t$ is the update gate, and $H_t$ is the memory state, with all of them being of the same dimensions as the hidden state matrix. $W_r, V_r, W_u, V_u, W_h, V_h \in \mathbb{R}^{(N+1) \times d \times d}$ are the hidden-to-hidden transition tensor, and $W_i, V_i \in \mathbb{R}^{(N+1) \times d \times 1}$ is the input-to-hidden transition tensor. The terms $W_r \circ H_{t-1}$ and $V_r \circ x_t$ capture the information from the previous hidden state and the newly coming input data, respectively. The operation $\circ$ signifies the product of two tensors along dimension $N + 1$, that is, $W_r \circ H_{t-1} = \begin{bmatrix} W^1_h h^1_{t-1}, \ldots, W^N_{h^{N+1}} h^{N+1}_{t-1} \end{bmatrix}$, where $W_r h^n_{t-1} \in \mathbb{R}^d$. The operation $\circ$ denotes element-wise multiplication. The produced hidden states are sent to the tandem attention layer to generate the context vector as the input to the ODE network.

### 3.3 Tandem Attention Mechanism

Two types of mechanisms are designed to dispatch temporal and variable attention and facilitate interpretability. Variable attention is generated based on the information distilled by temporal attention and hidden states considering the interaction between temporal and variable attention. Temporal attention is used to obtain the contributions in the time domain given the hidden states $[h^1_t, \ldots, h^{N+1}_t]$ corresponding to the $n$th input feature. The temporal attention weight $\alpha^n_i$ is computed as:

$$\alpha^n_i = \frac{\exp(f_a(h^n_i))}{\sum_s \exp(f_a(h^n_s))},$$

where $f_a(\cdot)$ is a feedforward neural network specific to the $n$th input feature. Since the designed tandem attention module is also tensorized, we use different linear projections to transform the tensorized hidden states for the computation of temporal attention. The scalar $\alpha^n_i$ controls the influence of the $n$th input feature at time step $t$. Subsequently, a variable-wise attention layer is used to generate the contributions in the feature domain based on the temporal attention weight and the hidden states. The variable attention weight is computed as:

$$\beta = \text{softmax} \left( f \left( \sum_i \alpha^n_i \circ H_i \right) \right),$$

where $f$ can be a feedforward neural network and $\beta \in \mathbb{R}^{N+1}$ controls the influence of each input feature for the prediction. Combining the two types of attention weights and hidden states, a context vector is generated through:

$$C_T = \sum_i \beta \circ \sum_j \alpha^n_j \circ (H_i),$$

where $\alpha^n_j$ is the hidden dimension per variable.

### 3.4 ODE Network

Following the processing of tandem attention, the context vector $C_T$ is fed to the ODE solver in order to provide outputs with continuous time intervals. A hidden state in a recurrent neural network decoder is transformed as:

$$h_{t+1} = h_t + f(h_t, \theta).$$

When adding more layers and taking smaller steps in Eq. (8), parameterized continuous dynamics of hidden units are obtained through an ODE module implemented as a neural network [26], based on:

$$\frac{dh(t)}{dt} = f(h(t), t, \theta).$$

In this form, the ODE network parameterizes the derivative of the hidden states rather than directly parameterizing the states themselves, and this leads to a continuous hierarchy rather than a discrete one. Furthermore, the mechanism has a constant memory cost without any need to store intermediate quantities of the forward pass during the training procedure. To get the numerical solution of $h(t_t)$, the network $f(h(t), t, \theta)$ can be integrated over the range $[t_0, t_1]$ with initial value $h(t_0)$, and the integral can be estimated through a black-box differential equation solver.

In arbitrary-step prediction, given the predicted time interval $\{T + m_1, \ldots, T + m_K\}$ and an initial latent value $z_T$, the ODE solver produces the estimated latent states $z_{T+m_1}, \ldots, z_{T+m_K}$. The ODE network operation is summarized as:

$$z_T \sim p(z_T),$$

$$[z_{T+m_1}, \ldots, z_{T+m_K}] = \text{ODESolver}(z_T, t, \theta, T + m_1, \ldots, T + m_K).$$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
In the ODE solver, represented in Fig. 4 (as a dashed rectangle), $f$ is a time-invariant function that takes the value $z_T$ at the current time step and outputs the gradient $\Delta z(t) = f(z(t), \theta_f)$, where $\theta_f$ is the parameter to be learned. In Eq. (10), we choose the basic GRU as the function $f$. With the property of the updated hidden state, the obtained difference equation is:

$$\Delta h_t = h_t - h_{t-1} = (1 - u_t) \odot h_{t-1} + u_t \odot \hat{h}_t - h_{t-1} = u_t \odot (\hat{h}_t - h_{t-1}).$$

$\Delta h_t$ represents the increment between the current and the previous hidden state, while $u_t$ and $h_t$ are the update gate and memory state of the GRU cell. The representation of ODE for $h(t)$ is obtained as:

$$\frac{dh(t)}{dt} = u_t \odot (\hat{h}_t - h_t).$$

As shown in Fig. 4, $p(z_T)$ is the prior distribution of $z_T$ assumed to be Gaussian and approximated with the posterior distribution $q(z_T|X_1 \cdots X_T)$. Using the input features, we infer the parameters for the posterior distribution over $z_T$:

$$q(z_T|C_T, \varphi) = \mathcal{N}(z_T|\mu_{z_T}, \sigma_{z_T}),$$

where $\varphi$ are parameters to learn and $(\mu_{z_T}, \sigma_{z_T})$ comes from the context vector $C_T$ generated from the TGRU and the tandem attention layers. Here we employ the re-parameterization trick proposed in [30] for the sampling process of $z_T$. Given the arbitrary prediction time interval, model predictions of the target series are obtained with the generated latent states $z$ through the ODE network:

$$\hat{y}_{T+m_1}, \ldots, \hat{y}_{T+m_K} = f_c(z_{T+m_1}, \ldots, z_{T+m_K}),$$

where $f_c$ represents the fully connected layer.

### 3.5 Loss Function

Since $C_T$ is generated from the TGRU and the tandem attention modules sampled from a Gaussian distribution, we refer to the standard Variational AutoEncoder (VAE) [30] algorithm to maximize the evidence lower bound (ELBO) which makes the model distribution approximate to the prior distribution in the latent space. Maximizing the ELBO can be transformed to minimizing the Kullback-Leibler (KL) divergence and maximizing the log-likelihood. To make the prediction results more accurate, the Mean Square Error (MSE) between the predicted and the true values is also considered.

Overall, the objective function here is configured with the following three parts, of which the first one is the MSE defined as:

$$\text{Loss}_{\text{MSE}} = \frac{1}{L \cdot K} \sum_{i=1}^{L} \sum_{k=1}^{K} (\hat{y}_i - y_i)^2,$$

where $L$ is the number of training samples and $K$ the number of predicted values. The second part is the KL-divergence between the prior $p(z_T)$ and the posterior distribution $q(z_T)$:

$$\text{Loss}_{\text{KL}} = \frac{1}{L} \sum_{i=1}^{L} D_{KL}(p(z_T)||q(z_T)),$$

aiming to bring them closer in the latent space. The final part is the negative log-likelihood of $y$ capturing the fitting degree of model parameters:

$$\text{Loss}_{\text{all}} = -\frac{1}{L} \sum_{i=1}^{L} \log p(y_i),$$

where $\log p(y_i)$ is the logarithmic likelihood. Model training is based on minimizing the overall loss function:

$$\text{Loss} = \text{Loss}_{\text{MSE}} + \text{Loss}_{\text{KL}} + \text{Loss}_{\text{all}}.$$

### 4 Experiment and Evaluation

In this section, we evaluate the effectiveness and stability of the proposed ETN-ODE by conducting arbitrary-step predictions with multivariate time series. Furthermore, we design five different standard multi-step prediction tasks compared against six deep learning methods, one non-deep learning method, and one ensemble-based method. To demonstrate the efficiency of ETN-ODE, we also perform an ablation study and include visualization of the tandem attention mechanism. Our code is available at https://github.com/PengleiGao/ETN-ODE.

#### 4.1 Datasets

The following five real-world datasets are used in our experiments:

- **SML2010** [31]: It is a public dataset used for indoor temperature forecasting sampled every minute. The room temperature is taken as the target series and another 13 time series are exogenous variables containing approximately 40 days of monitoring data.
- **PM2.5** [32]: This is a public dataset from the UCI repository containing hourly recorded PM2.5 data in Beijing, China. We use PM2.5 measurements as the target series and another 6 time series as the exogenous variables.
- **Energy** [33]: This dataset from the UCI repository consists of 25 exogenous variables recorded every 10 minutes for 4.5 months. We choose the appliances energy consumption series as the target series.
- **Nasdaq100** [7]: In this public dataset, stock prices from 81 major corporations are selected as the exogenous variables recorded every minute, and the Nasdaq100 index value is used as the target.
- **Electricity** [34]: This is a public dataset for electricity consumption prediction of Homestead, US. The consumption is chosen as the target series sampled hourly, while the other 15 time series are exogenous variables containing weather features.

#### 4.2 Experimental Settings and Metrics

For all datasets, the time window size $T$ for input data is set to 20. The size of the hidden layer in our model and the IMV-tensor model is set by the number of neurons per variable from $\{5, 10, 15, 20, 25\}$. In the VAR model, the order of autoregression is set to the window size. For the XGB model, the
This method combines the exponential and mean absolute error:

\[
\text{Mean Absolute Error: } \frac{1}{p} \sum_{t=1}^{p} |\hat{y}_t - y_t|.
\]

4.3 Baselines

The methods we employ for our comparisons are:

VAR [35]: The popular Vector Autoregression model for multivariate time series forecasting.

XGB [36]: The ensemble-based method Extreme Gradient Boosting which is an application of boosting with regression trees.

ES-RNN [37]: This method combines the exponential smoothing and recurrent neural networks for time series forecasting.

Retain [23]: This model uses two types of attention generated with two RNNs for classification with reversed time inputs. We replace the output layer with a fully connected layer for multi-step prediction.

UA [14]: This model uses variational inference on the attention part to learn the model uncertainty based on Retain. We replace the output layer with a decoder layer to adapt it for multi-step prediction.

IMV-tensor [13]: It employs a tensorized LSTM to capture different dynamics in multivariate time series and mixture attention to model the generative process of the target series for the next value prediction.

TLSTM [20]: It uses higher-order moments and state transition functions in LSTM for long-term forecasting, using the encoder-decoder structure.

Latent-ODE [26]: It uses a variational autoencoder structure with reversed time inputs, with both encoder and decoder networks being RNNs.

4.4 Evaluation Results

4.4.1 Arbitrary-Step Prediction

As very few studies have focused on multivariate time series for arbitrary-step prediction through building continuous networks, we only evaluate the effectiveness of arbitrary-step prediction of the proposed model against the Latent-ODE. The aim is to forecast multiple future values that are not sampled in the original time series by setting continuous time intervals in the testing stage based on Eq. (10). The forecasting interval can be different between each neighboring predicted point, while fixed for standard multi-step prediction. To better demonstrate the arbitrary-step prediction quantitatively, we resample the dataset to half of its original size by taking twice the sampling gap. During training, the model only outputs three future values at discrete time points sharing the same sample gap as the input data, i.e., \( T + 1, T + 2, T + 3 \). In the testing stage, the model outputs two extra future values at continuous steps, e.g., \( T + 1.5 \) and \( T + 2.5 \), which are not involved with training. For instance, after resampling the energy dataset, we can forecast the values of the next ‘thirty-minute’ and ‘fifty-minute’ steps, when the data is sampled every twenty minutes.

Table 1 shows the forecasting errors of each predicted step for the arbitrary-step prediction task. The results show that ETN-ODE can significantly outperform Latent-ODE on RMSE and MAE metrics due to the adaptability of the TGRU and the tandem attention mechanism. In the Latent-ODE model, the function \( f \) in Eq. (10) is chosen to be a fully-connected network with two layers. Instead, for the ETN-ODE architecture we use a basic GRU network to model the incremental relationship between each neighboring time point. As discussed in section 3.4, the transformation of hidden states in the GRU leads to a proper ODE formulation which enhances the performance of arbitrary-step prediction.

| Dataset       | SML2010 | Energy | PM2.5 | Nasdaq100 | Electricity |
|---------------|---------|--------|-------|-----------|-------------|
|               | RMSE    | MAE    | RMSE  | MAE       | RMSE        | MAE         |
| Step1         | 0.099   | 0.079  | 76.552| 36.829    | 29.661      | 18.127      |
| Step1.5       | 0.097   | 0.077  | 81.948| 40.793    | 36.942      | 22.522      |
| ETN-ODE       | 0.106   | 0.082  | 80.259| 40.658    | 43.284      | 26.427      |
| Step2         | 0.118   | 0.089  | 84.237| 42.425    | 48.159      | 29.803      |
| Step2.5       | 0.139   | 0.105  | 82.531| 42.301    | 52.947      | 33.180      |
| Step1         | 0.647   | 0.557  | 84.134| 43.871    | 51.600      | 38.759      |
| Step1.5       | 0.681   | 0.588  | 87.206| 45.032    | 55.085      | 40.658      |
| Latent-ODE    | 0.718   | 0.621  | 85.532| 45.041    | 61.916      | 35.015      |
| Step2         | 0.757   | 0.653  | 88.748| 45.988    | 65.391      | 47.731      |
| Step2.5       | 0.799   | 0.685  | 86.993| 45.907    | 65.391      | 47.731      |

TABLE 1

Results of Arbitrary-Step Prediction, Where Lower Values Correspond to Better Performance
Step1.5 and Step2.5 achieve forecasting errors similar to the other three steps demonstrating the effectiveness of our model for arbitrary-step prediction. In Figs. 5 and 6, we further visualize the predicted values of the target series for ‘Step1.5’ and ‘Step2.5’ on SML2010 and Electricity datasets. The red dashed line describes the predicted values of ETN-ODE which captures the period information perfectly and fits much better to the target series than Latent-ODE, as shown in both figures. Also, ETN-ODE performs much better than Latent-ODE on Electricity which has a faster fluctuation rate, and this result indicates that our model can handle datasets with different volatility. Additionally, it is noted that on Energy both models lead to relatively larger errors which can be partially explained by the dataset’s complex properties and noise. Nonetheless, ETN-ODE still attains better performance than Latent-ODE.

In Fig. 7, we can visualize the variable attention for arbitrary-step prediction and qualitatively analyze the meaningfulness and effectiveness of the variable contribution in the tandem attention module. The variable contribution values converge during the training and are determined at the end of the procedure. Variables with higher ranking contribute more to the target series. For the SML2010 dataset in Fig. 7a, the variable ‘Sun dusk’ has the most contribution to the prediction during the training stage, indicating that the room temperature will rise when the sun is shining brightly. For the Electricity dataset in Fig. 7b, the variables ‘wind’, ‘temperature’, and ‘cloud’ contribute more to the target series. When the temperature is very high or low, the power consumption will increase, which is consistent with actual situations. These results show that the variable attention can allocate appropriate contributions when evaluated on the arbitrary-step prediction.

4.4.2 Standard Multi-Step Prediction

We evaluate the different methods on five standard multi-step prediction tasks through forecasting the next 1, 2, 5, 8, and 10 future values of the target series. Tables 2, 3, 4, 5, and 6 present the RMSE and MAE errors for all methods on...
the five datasets (best performances are shown boldfaced and second-best underlined). The results of the proposed ETN-ODE are listed in the bottom row of each table section for each evaluation metric. It can be observed that ETN-ODE achieves either the best or the second-best performance on all the tasks with respect to RMSE and MAE on SML2010, PM2.5, Electricity, and Energy datasets, and competitive results on the Nasdaq100 dataset.

In particular, ETN-ODE achieves significant improvements on SML2010 and Electricity over the multivariate forecasting model IMV-tensor, generating on average an 18.65% and 5.19% decrease in RMSE and 19.63% and 6.57% decrease in MAE on the two datasets, respectively. Also, the proposed method shows better performance on all datasets compared to the ensemble-based XGB model and is comparable to the deep learning methods.

Compared to the non-deep learning VAR, ETN-ODE achieves better performance except for the Nasdaq100 dataset. Similarly, the deep learning methods Retain, UA, and ES-RNN perform better than ETN-ODE on the Nasdaq100 dataset. A possible reason is that there are 81 exogenous variables in Nasdaq100, which contain much redundant information. It appears that most of the exogenous variables are not very correlated and have limited effect on prediction, which could also be observed from the visualization of the variable contributions in Fig. 9e. Our method allocates different importance to each individual exogenous variable and when in this dataset the index price mainly depends on historical data of itself, some uncorrelated exogenous variables may cause negative effects.

Moreover, ETN-ODE outperforms ES-RNN on the PM2.5 dataset for all five standard multi-step prediction tasks.
while ES-RNN performs better on the Nasdaq100 dataset. ETN-ODE and ES-RNN achieve similar results on other datasets while the former takes advantage of long-range forecasting. For Nasdaq100, ES-RNN only takes the historical data of the index price as input with less redundant information which seems to perform well on this low-frequency data. For long-term period prediction, ETN-ODE has the advantage of modeling the long-range dependencies through full use of the exogenous variables. Allocating different degrees of importance to these variables provides extra information for long-rang predictions when only considering the historical data of the target series. Importantly, our proposed continuous-time model can forecast future values at arbitrary time points and provide interpretability for the exogenous variables.

ETN-ODE outperforms significantly the Latent-ODE which is another continuous-time model. The hidden states of the latter are generated with a basic RNN which bears limitations in providing appropriate mappings to the ODE network when processing multivariate time series. This corroborates the design choice of ETN-ODE having a TGRU layer to produce informative hidden states that represent different dynamics specific to each input feature. Moreover, the tandem attention layer helps to allocate different contributions to the predictions enhancing thus the predictive ability.

Table 7 summarizes the main results from the above experimental comparison. Overall, it can be seen that while ETN-ODE is designed mainly for continuous predictions, it can also demonstrate outstanding performance on standard

### TABLE 4

| Dataset          | Metric | Method       | M=1    | M=2    | M=5    | M=8    | M=10   |
|------------------|--------|--------------|--------|--------|--------|--------|--------|
|                  |        |              | 4.091  | 6.767  | 23.044 | 69.575 | 0.076  |
|                  |        |              | 3.043  | 2.422  | 1.373  | 0.977  | 0.042  |

**TABLE 5**

| Dataset          | Metric | Method       | M=1    | M=2    | M=5    | M=8    | M=10   |
|------------------|--------|--------------|--------|--------|--------|--------|--------|
|                  |        |              |        |        |        |        |        |
multi-step time series prediction. This advantage stems from its better representation capabilities through the parameterization of the latent state derivatives.

4.4.3 Ablation Study

We select the following four ablation study variants to demonstrate the effectiveness of our model components:

- **TLSTM-ODE**: Replace the TGRU with tensorized LSTM in the ETN-ODE framework.
- **w/oODE**: Remove the ODE network component such that the model is no longer a continuous network. We then use a fully connected layer to output the predictions.
- **w/oATT**: Remove the tandem attention layer and use mean pooling operated on the hidden states produced by the TGRU layer to generate the context vector \( C_T \).
- **ODE-va**: Extract the variable attention \( \beta \) in the tandem attention layer and obtain the prediction with a weighted sum of the variable attention \( \beta \) and the latent states \( z \).

Selecting as a case study the task of forecasting 5 future values in standard multi-step prediction, we present our ablation study results through Fig. 8. In general, with all its necessary components ETN-ODE achieves the best performance consistently on the five datasets and on both metrics, whereas removing components decreases performance. The difference between each model variant is not apparent for the Energy and PM2.5 datasets. The similar behavior is possibly caused by the target series of the two datasets containing a lot of high-frequency information where the time series contain steep variations over short time steps. This leads to a difficulty in networks learning the underlying high-frequency dynamical behavior, and the capability of the ODE network cannot be fully utilized. However, for the other datasets, it is not straightforward to establish whether each component is relevant to the model. Some notable observations to be highlighted here are:

- Without the ODE module, the model cannot output future values at arbitrary time points. Additionally, it is efficient to model the gradient of the latent states in order to improve performance.
- The TGRU appears more useful in attaining higher performance than the tensorized LSTM. We attribute this to its better representation capabilities through the parameterization of the latent state derivatives.

| Method | #best results of total tasks | #2nd best results of total tasks | #best results (M=8,10) for long-range forecasting | #2nd best results (M=8,10) for long-range forecasting | #best results of total tasks | capable of arbitrary-step prediction |
|--------|------------------------------|---------------------------------|-----------------------------------------------|-----------------------------------------------|-----------------------------|----------------------------------|
| ETN-ODE | 20                           | 20                              | 4                                             | 4                                             | 50                          | ✓                               |
| ES-RNN  | 31                           | 7                               | 9                                             | 5                                             | -                           | ×                               |
| IMV-tensor | 0                     | 12                              | 0                                             | 6                                             | -                           | ×                               |
| VAR     | 0                            | 8                               | 0                                             | 4                                             | -                           | ×                               |
| Retain  | 0                            | 2                               | 0                                             | 0                                             | -                           | ×                               |
| Latent-ODE | 0                     | 0                               | 0                                             | 0                                             | 0                           | ✓                               |

### TABLE 6

| Dataset | Metric | Method | Energy |
|---------|--------|--------|--------|
|         |        |        | MAE    |
|         |        |        | RMSE   |
| VAR     | M=1    | 62.850±0.242 | 20.510±0.977 |
| XGB     | M=2    | 70.328±0.063 | 24.865±1.097 |
| ES-RNN  | M=5    | 77.706±0.066 | 28.760±1.403 |
| TLSTM   | M=8    | 80.795±0.143 | 32.588±1.705 |
| Latent-ODE | M=10 | 81.789±0.176 | 36.221±2.045 |
| MAE     | M=1    | 31.598±0.418 | 14.384±0.973 |
| XGB     | M=2    | 34.325±0.878 | 18.030±1.345 |
| ES-RNN  | M=5    | 37.420±1.807 | 20.365±1.877 |
| TLSTM   | M=8    | 41.519±0.695 | 24.865±1.097 |
| Latent-ODE | M=10 | 45.848±0.902 | 28.760±1.403 |

### TABLE 7

Summary of Comparisons on the Number of Best and Second Best Results for Several Comparative Methods (The Number of Total Tasks is 50 for Both Standard Multi-Step and Arbitrary-Step Prediction)
this superiority to its fewer parameters, leading to faster convergence and better performance.

- The attention component substantially affects the prediction performance which is critical in ETN-ODE. The employed tandem attention mechanism can lead to interpretability which will be visualized in the next section.
- Compared with the ODE-va, the way of sampling the context vector $C_T$ as described in Section 3, also affects the performance. Generating $C_T$ with both $\alpha$ and $\beta$ in the tandem attention layer can achieve better performance, and this provides a better input to the ODE network. Adding variable attention $\beta$ after the ODE network hardly affects the model performance while modeling the gradient of the latent states.

4.4.4 Visualization of the Explanation for Exogenous Variables

With the tandem attention mechanism of ETN-ODE, both the variable and temporal contributions can be visualized and these offer insight in the interpretability of the model. The variable contributions can explain the importance of each exogenous variable shown in Fig. 9 which demonstrate the effectiveness of the attention mechanism where variables with higher ranking contribute more to the prediction during training. Similar to Fig. 7, the variable contributions converge during the training and are determined at the end of training. Specifically, in Fig. 9a, for the SML2010 dataset, the dining room temperature and carbon dioxide content have the most significant impact to the predictions. The temperature rises when people cook in the kitchen. In Fig. 9b, for the PM2.5 dataset, we can observe that variables ‘Rain’, ‘Wind speed’ and ‘Autoregressive’ have high contributions, while the variable ‘Temperature’ has the lowest contribution. Strong winds can bring dry and fresh air, while rain can settle air pollutants. As shown in Fig. 9c for the Electricity dataset, variables ‘DewPoint’, ‘winddirDegree’, and ‘Pressure’ contribute more to the predictions. Indeed, a decrease in the dew point and an increase in the pressure are usually
accompanied by the arrival of the northerly wind, which can influence the electricity consumption. In Fig. 9e, variables ‘DewPoint’, ‘Visibility’, and ‘T_7’ are the three most important variables contributing to the prediction. Variable ‘T_7’ represents the temperature in ironing room, where iron is a high energy consumption appliance. Variables ‘DewPoint’ and ‘Visibility’ influence the weather conditions that are more related to appliance energy consumption. The three variables are highly correlated with the target series consistent with actual situations. In Fig. 9e, the index price itself, variable ‘STX’ and variable ‘JD’ contribute most to the index price movements.

The contribution visualization of Fig. 9 slightly differs from the results of Fig. 7 due to the different input sampling rates between arbitrary-step and standard multi-step prediction, which leads to a change in the data frequency that reduces the frequency of effective information. As such, the network learns different behavior leading to a difference between the two experiments. Variables contribute more to the target series with high-frequency data but may contribute less with low-frequency data. The tandem attention layer can also explain the variable contributions in arbitrary-step prediction. The variable attention has a high fluctuation on Nasdaq100 and a low fluctuation on Energy during training, as shown in Figs. 9e and 9d. For Nasdaq100, after epoch 20, the variable contributions begin to take effect on the predictions, indicating that the tandem attention layer works sufficiently well.

In Figs. 10, 11, 12, 13, and 14, we demonstrate the temporal attention values at the end of training for the five datasets on short/long-term forecasting periods (K = 1 or 10), where lighter colors indicate that the corresponding data contributes more to the predictions. Specifically, the short history of the variables ‘Autoregressive’ and ‘Temp.outdoor’ contributes more to the short-term forecasting period while lighting and sun irradiance influence more the long-term forecasting period on SML2010 in Fig. 10. We can also observe that the temporal attention tends to allocate more weight to the long-historical data of the long-term forecasting period task. On PM2.5 in Fig. 11, the variables ‘Rain’ and ‘Autoregressive’ have higher contribution to the short-term forecasting period with their short historical information, while on the long-term forecasting period the variable ‘Snow’ has a relatively higher contribution with its long

Fig. 10. Temporal contribution on SML2010 for short/long-term forecasting (K = 1 or 10) at the end of training.

Fig. 11. Temporal contribution on PM2.5 for short/long-term forecasting (K = 1 or 10) at the end of training.

Fig. 12. Temporal contribution on Electricity for short/long-term forecasting (K = 1 or 10) at the end of training.

Fig. 13. Temporal contribution on Energy for short/long-term forecasting (K = 1 or 10) at the end of training.

Fig. 14. Temporal contribution on Nasdaq100 for short/long-term forecasting (K = 1 or 10) at the end of training.
historical information. For the Electricity dataset in Fig. 12, ‘windspeed’ contributes more with its short historical information while the variable ‘winddirDegree’ has little influence with its short historical information in the long-term forecasting period. For Energy and Nasdaq100, similar results can be seen in Figs. 13 and 14, where the temporal attention allocates different temporal contributions on different forecasting period tasks and tends to allocate more weights to the long-historical data of the long-term tasks.

4.4.5 Parameter Sensitivity

Finally, we study the parameter sensitivity of ETN-ODE with respect to the noise standard deviation in $\text{Loss}_{\text{std}}$, the hidden dimension per variable in the TGRU layer, and the loss function penalty, with results presented in Fig. 15. We scale the RMSE error for better visualization on task $K = 5$. For SML2010, the model performance declines only when noise is greater than 0.05. For Electricity, the RMSE error declines notably when the noise is greater than 0.1. For Nasdaq100, prediction is relatively sensitive to the hyperparameters mainly due to the large number of input features. For Energy and PM2.5 these parameters have a limited influence on model performance. Overall, our model does not show noteworthy sensitivity to hyperparameters.

5 Conclusion & Future Work

In this work, we propose the explainable continuous framework ETN-ODE for arbitrary-step prediction with multiple variables which can output arbitrary predicted values via utilizing an ODE network. We design a TGRU model to process the multivariate time series representing different dynamics of individual input features with fewer parameters to be learned in the network. A tandem attention mechanism is also proposed to generate more adaptive inputs to the ODE network, providing interpretability by visualizing the temporal and variable contributions. Various experiments on arbitrary-step prediction and standard multi-step prediction on multiple real-world datasets demonstrate the effectiveness of our model.

Future work will include modification of this continuous model for time series sampled at random times. For instance, in medical data analysis data on patient visits are typically sampled irregularly and are recorded at different times without all health indicators being fully registered every time. Consequently, a suitable architecture could be one with an ODE module equipped with encoding and decoding adapting to the random input length and arbitrary future values. Another useful scenario would be to deal with missing values. A continuous model could interpolate the dataset and predict the future values at other time instants based on relationships between input and target series given partial observations at certain fixed times. Moreover, adding an attention mechanism internally within the ODE solver would be a promising mechanism for improving further model performance and enhancing interpretability.

REFERENCES

[1] J. Cheng, K. Huang, and Z. Zheng, “Towards better forecasting by fusing near and distant future visions,” in Proc. 34th Assoc. Adv. Learn. Theory. Conf. Artif. Intell., 2020, pp. 3593–3600.
[2] J. Wang, T. Sun, B. Liu, Y. Cao, and H. Zhu, “CLVSA: A convolutional LSTM based variational sequence-to-sequence model with attention for predicting trends of financial markets,” in Proc. 28th Int. Joint Conf. Artif. Intell., 2019, pp. 3705–3711.
[3] K. Zhao, D. Khryashev, and H. Vo, “Predicting taxi and uber demand in cities: Approaching the limit of predictability,” IEEE Trans. Knowl. Data Eng., vol. 33, no. 6, pp. 2723–2736, Jun. 2021.
[4] Y. Zhang et al., “Multi-group encoder-decoder networks to fuse heterogeneous data for next-day air quality prediction,” in Proc. 28th Int. Joint Conf. Artif. Intell., 2019, pp. 4341–4347.
[5] S. Du, T. Li, Y. Yang, and S. Horng, “Deep air quality forecasting using hybrid deep learning framework,” IEEE Trans. Knowl. Data Eng., vol. 33, no. 6, pp. 2412–2424, Jun. 2021.
[6] S. Masum, Y. Liu, and J. Chiverton, “Multi-step time series forecasting of electric load using machine learning models,” in Proc. 17th Int. Conf. Artif. Intell. Soft Comput., 2018, pp. 148–159.
[7] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, “A dual-stage attention-based recurrent neural network for time series prediction,” in Proc. 26th Int. Joint Conf. Artif. Intell., 2017, pp. 2627–2633.
[8] L. Li, J. Yan, X. Yang, and Y. Jin, “Learning interpretable deep state space model for probabilistic time series forecasting,” in Proc. 28th Int. Joint Conf. Artif. Intell., 2019, pp. 2901–2908.
[9] S. Du, T. Li, Y. Yang, and S.-J. Horng, “Multivariate time series forecasting via attention-based encoder–decoder framework,” Neurocomputing, vol. 388, pp. 269–279, 2020.
[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” Nature, vol. 323, no. 6088, pp. 533–536, 1986.
[11] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[12] K. Cho et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in Proc. Conf. Empir. Methods Natural Lang. Process., 2014, pp. 1724–1734.
[13] T. Guo, T. Lin, and N. Antulov-Fantulin, “Exploring interpretable LSTM neural networks over multi-variable data,” in Proc. 36th Int. Conf. Mach. Learn., vol. 97, 2019, pp. 2494–2504.
[14] J. Heo et al., “Uncertainty-aware attention for reliable interpretation and prediction,” in Proc. 31st Natl Informal. Process. Syst., 2018, pp. 909–918.
[15] H. Akaike, “Fitting autoregressive models for prediction,” Ann. Inst. Statist. Math., vol. 21, no. 1, pp. 243–247, 1969.
[16] C. A. Sims, “Macroeconomics and reality,” Econometrica: J. Econometric Soc., vol. 48, pp. 1–48, 1980.
[17] K. Huang, A. Hussain, Q.-F. Wang, and R. Zhang, Deep Learning: Fundamentals Theory Applications. Berlin, Germany: Springer, 2019.
[18] H. Yang, K. Huang, I. King, and M. R. Lyu, “Localized support vector regression for time series prediction,” Neurocomputing, vol. 72, no. 10–12, pp. 2659–2669, 2009.

[19] I. Fox, L. Ang, M. Jaiswal, R. Pop-Busui, and J. Wiens, “Deep multi-output forecasting: Learning to accurately predict blood glucose trajectories,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2018, pp. 1387–1395.

[20] R. Yu, S. Zheng, A. Anandkumar, and Y. Yue, “Long-term forecasting using higher order tensor RNNs,” 2017, arXiv:1711.00073.

[21] Z. Yang, W. Yan, X. Huang, and L. Mei, “Adaptive temporal-frequency network for time-series forecasting,” IEEE Trans. Knowl. Data Eng., vol. 34, no. 4, pp. 1576–1587, Apr. 2022.

[22] C. Zang, “Deep learning in multiple multistep time series prediction,” CoRR, 2017, arXiv:1710.04371.

[23] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, “Rretain: An interpretable predictive model for healthcare using reverse time attention mechanism,” in Proc. 29th Neural Inform. Process. Syst., 2016, pp. 3504–3512.

[24] Z. He, S. Gao, L. Xiao, D. Liu, H. He, and D. Barber, “Wider and deeper, cheaper and faster: Tensorized LSTMs for sequence learning,” in Proc. 30th Neural Inform. Process. Syst., 2017, pp. 1–11.

[25] D. Xu et al., “Tensorized LSTM with adaptive shared memory for learning trends in multivariate time series,” in Proc. 34th Assoc. Adv. Artif. Intell., 2020, pp. 1395–1402.

[26] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” in Proc. 31st Neural Inform. Process. Syst., 2018, pp. 6571–6583.

[27] Y. Rubanova, R. T. Chen, and D. Duvenaud, “Latent ODEs for irregularly-sampled time series,” CoRR, 2019, arXiv:1907.03907.

[28] T. Asakis, L. Böttcher, and N. Antulov-Fantulin, “Neural ordinary differential equation control of dynamics on graphs,” 2020, arXiv:2006.09773.

[29] E. De Brouwer, J. Simm, A. Arany, and Y. Moreau, “GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series,” in Proc. 32nd Neural Inform. Process. Syst., 2019, pp. 7377–7388.

[30] D. P. Kingma and M. Welling, “Auto-Encoding variational Bayes,” in Proc. 2nd Int. Conf. Learn. Representations, 2014.

[31] F. Zamora-Martinez, P. Romeu, P. Botella-Rocamora, and J. Pardo, “On-line learning of indoor temperature forecasting models towards energy efficiency,” Energy Buildings, vol. 83, pp. 162–172, 2014.

[32] X. Liang et al., “Assessing beijing’s PM2.5 pollution: Severity, weather impact, apec and winter heating,” Roy. Soc. A: Math. Phys. Eng. Sci., vol. 471, no. 2182, 2015, Art. no. 20150257.

[33] L. M. Candanedo, V. Feldheim, and D. Deramaix, “Data driven prediction models of energy use of appliances in a low-energy house,” Energy Buildings, vol. 140, pp. 81–97, 2017.

[34] A. Jain, “Homestead/us electricity consumption,” 2020. [Online]. Available: https://www.kaggle.com/anujtheb/homestead-us-electricity-consumption

[35] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control. Hoboken, NJ, USA: Wiley, 2015.

[36] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 785–794.

[37] S. Smyl, “A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting,” Int. J. Forecasting, vol. 36, no. 1, pp. 75–85, 2020.

Penglei Gao received the BSc degree in applied statistics from the Shanghai University of Finance and Economics (SUFE), Shanghai, China, in 2015, and the MSc degree in financial mathematics from Xi’an Jiaotong-Liverpool University (XJTLU), in 2017. He is currently working toward a PhD degree with the University of Liverpool (UoL), U.K. His research interests include time series and deep learning.

Xi Yang received the BSc degree in mathematics from Finance, the MSc degree with distinction in computer science, and the PhD degree in engineering from the University of Liverpool, Liverpool, U.K. She is currently an assistant professor at Xi’an Jiaotong-Liverpool University. Her research interest includes the field of Bayesian probabilistic approaches to machine learning, pattern recognition and data mining.

Rui Zhang received the first-class (Hons) degree in telecommunications engineering from the Jilin University of China, in 2001, and the PhD degree in computer science and mathematics from the University of Ulster, U.K., in 2007. After finishing her PhD, she joined Xi’an Jiaotong-Liverpool University and currently holds the position of associate professor. Her research interests include machine learning, data mining, and statistical analysis.

Kaizhu Huang received the PhD degree from the Chinese University of Hong Kong (CUHK), in 2004. His research interests include machine learning, neural information processing, and pattern recognition. He is now a full professor with the Institute of Applied Physical Sciences and Engineering, Duke Kunshan University. He was a full professor with the Department of Intelligent Science, Xian Jiaotong-Liverpool University (XJTLU) and associate dean of research with the School of Advanced Technology, XJTLU. He founded Suzhou Municipal Key Laboratory of Cognitive Computation and Applied Technology. He worked in Fujitsu Research Centre, CUHK, University of Bristol, National Laboratory of Pattern Recognition, Chinese Academy of Sciences from 2004 to 2012. He was the recipient of the 2011 Asia Pacific Neural Network Society Young Researcher Award. He received Best Paper or Book Award eight times. He serves as corresponding editors/advisory board members eight journals and book series. He invited as keynote speaker in more than 30 international conferences or workshops.

John Y. Goulermas received the BSc (1st class) degree in computation from the University of Manchester (UMIST), in 1994, and the MSc and PhD degrees from the Control Systems Center, University of Manchester, in 1996 and 2000, respectively. He is currently a professor with the Department of Computer Science, University of Liverpool. His research interests include mathematical modeling, data analytics, and machine learning. He has worked on various applications including image analysis, biomedical engineering, industrial monitoring, and security.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.