Neurocomputing 468 (2022) 123–136

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Hybrid interpretable predictive machine learning model for air pollution prediction

Yuanlin Gu [a], Baihua Li [b], Qinggang Meng [c,*]

[a] Department of Computer Science, University of Roehampton, London, SW15 5PU, United Kingdom
[b] Department of Computer Science, Loughborough University, Loughborough, LE11 3TU, United Kingdom
[c] Department of Computer Science, Loughborough University, Loughborough, LE11 3TU, United Kingdom

ARTICLE INFO

Article history:
Received 15 March 2021
Revised 9 July 2021
Accepted 25 September 2021
Available online 6 October 2021
Communicated by Zidong Wang

Keywords:
Air pollution prediction
Interpretable machine learning
Neural network
NARMAX model

ABSTRACT

Air pollution prediction is a burning issue, as pollutants can harm human health. Traditional machine learning models usually aim to improve the overall prediction accuracy but neglect the accuracy for peak values. Moreover, these models are not interpretable. They fail to explain the interactions between various determining factors and their impacts on air pollution. In this paper, we propose a new Hybrid Interpretable Predictive Machine Learning model for the Particulate Matter 2.5 prediction, which carries two novelties. First, a hybrid model structure is constructed with deep neural network and Nonlinear Auto Regressive Moving Average with Exogenous Input model. Second, automatic feature generation and feature selection procedures are integrated into this hybrid model. The experimental results demonstrate the superiority of our model over other models in prediction accuracy for peak values and model interpretability. The proposed model reveals how PM2.5 prediction is estimated by historical PM2.5, weather, and season. The accuracies (measured by correlation coefficients) of 1, 3 and 6-hour-ahead prediction are 0.9870, 0.9332 and 0.8587, respectively. More importantly, the proposed approach presents a new interpretable machine learning framework for time series data, enabling to explain complex dependence of multimode inputs, and to build reliable predictive models.

1. Introduction

Air pollution has become a leading threat to public health worldwide. Millions of people are experiencing harms from the polluted air on a continuing basis [1]. In addition to the striking consequence on human health, air pollution also adversely affects the national economy as well as welfare [2]. In China, the air pollution is particularly problematic due to the rapid industrialization in recent years. Despite the ongoing "war on the pollution" carried out by the government, the air pollution level in China is still far beyond the healthy air-quality standard. In 2019, the average concentration of $PM_{2.5}$ in Beijing, the Chinese capital, is 42 μg per cubic meter, quadrupled the World Health Organization (WHO) recommendation [3]. Therefore, to create a cleaner and healthier environment, accurately predicting the pollution peaks and identifying the pollution determinants are of importance for the sake of establishing efficient pollution mitigation policies.

$PM_{2.5}$, which measures mass per cubic meter of air of particles with a size generally less than 2.5 μm, is one of the most important air pollution indices. Particularly, a high value of $PM_{2.5}$ indicates that the pollutants are extremely dangerous to human and may cause serious health problems such as pulmonary tuberculosis (TB) [4] and upper respiratory tract infections (URI) [5]. In air pollution studies, many data-driven models have been widely applied in PM2.5 forecasting, including feedforward neural networks [6–11], recurrent neural networks such as long short-term memory (LSTM) [12–17], Markov models [18,19], ensemble models [20–22], fuzzy models [23], Bayesian models [24,25], and regression-based methods [26–28].

Although some neural networks are popular in modelling $PM_{2.5}$ and have achieved the state-of-the-art prediction accuracy, they have two limitations. The first limitation is that they are not interpretable. A significant amount of domain knowledge and time is required to define suitable hand-crafted features and train traditional neural networks [14]. The procedure could also be prone to human errors, as prediction can be affected by a huge range of factors and many uncertain conditions. Consequently, redundant features might be included in the model, while important features may be omitted. Recent deep neural networks allow automatic fea-

[*] Corresponding author.
E-mail addresses: yuanlin.gu@roehampton.ac.uk (Y. Gu), b.li@lboro.ac.uk (B. Li), q.meng@lboro.ac.uk (Q. Meng).

https://doi.org/10.1016/j.neucom.2021.09.051
0925-2312/© 2021 Elsevier B.V. All rights reserved.

ture learning from large datasets and incidents, relying less on human operation and domain knowledge. Derived features are built by feature extraction through complex high-dimensional non-linear computation procedures. These derived features may facilitate the subsequent learning and generalization steps. However, the features and network structure of such neural networks are uninformative to humans, as they are not apparently related to the physical dimension and domain meaning of the inputs, making it extremely difficult to identify the important and useful features. If all the derived features are included in the model, the model can be overfitted, and the computation time will be largely increased. In addition, if some important input explanatory variables can vary or interact very differently at complex or extreme environmental conditions (e.g., peak PM2.5 periods), the model may fail to catch the changing dynamics of the system and become unreliable. At these situations, an interpretable prediction model and explainable features can reveal the relationship between system output and complex changing impact of inputs, making it easier to fine-tune the model with knowledge and insights gained. The second limitation is that traditional neural networks usually focus on the improvement of overall prediction accuracy, but neglect the prediction accuracy for peak values, the most serious pollution periods which could last for a long time and pose the most harms to the public. This is because the real data is usually imbalanced, characterized by much more normal values than peak values, thus makes it challenging to achieve good accuracy for peak values. A small number of key features usually have significant effects on peak values, and the identification of them are the keys to improve peak value prediction. Based on the above reasons, the interpretability of machine learning features, model dependence and performance are highly desirable in many real-world applications, and an open challenge faced by the machine learning community. Answers and insights to this question are crucial, as this will ultimately determine the reliability of a machine learning model to changing situations and its adaptability to possible new domains.

To overcome the above limitations, we propose a new Hybrid Interpretable Predictive Machine Learning (HIP-ML) model for $PM_{2.5}$ forecasting. The proposed model carries two novelties. First, we combine the artificial deep neural network with the interpretable Nonlinear Auto Regressive Moving Average with Exogenous Input (NARMAX) model to propose a new interpretable hybrid model structure [29]. Second, we incorporate both automatic feature generation and feature selection into this proposed model. The NARMAX model provides a feature selection process to derive effective time-lagged nonlinear features which can well describe the time dependences between output and inputs. Because the the time dependencies are well analyzed, the integrated deep neural network is mainly constructed by a number of dense layers and a Relu layer. This will largely reduce the computation cost for training the model. This paper makes several main contributions as follows.

1) This paper proposes the HIP-ML model, which provides an interpretable structure with integrated deep neural network. The proposed model automatically generates features and then selects the most important features based on their contributions. With the improved model interpretability, the model provides valuable information for air pollution prediction. Weather and seasonal factors (i.e., pressure, northwest wind, winter, sunlight) and their interactions are found to have significant impacts on air pollution prediction.
2) Concerning the complex and dynamic nature of $PM_{2.5}$, this article utilizes the proposed HIP-ML model, which uses a two-stage hybrid structure to reduce the prediction error of peak values, in $PM_{2.5}$ forecasting. The experimental results show that the proposed model is superior to traditional methods in overall prediction accuracy and prediction accuracy for peak values.

The paper is organized as follows. Section 2 presents the related works. The proposed method is introduced in Section 3. The experimental results and discussion are in Section 4, and finally the work is concluded in Section 5.

2. Related work

Some early-stage neural networks were built for air pollution prediction. Perez et al. [6] designed a neural network to predict 6-hour-ahead $PM_{2.5}$ in Santiago, Chile in 2011, and further improved the accuracy in 2001 [7]. Ordieres et al. [8] aimed to predict $PM_{2.5}$ forecast on the US-Mexico border in 2005. Benefit from the revolution of computation power and advances in neural network, some advanced neural networks (e.g., deep LSTM) were applied to predict $PM_{2.5}$ in recent years. In 2019, Bai et al. [16] built a LSTM for one-hour-ahead $PM_{2.5}$ prediction in China, with RMSE of 13–14 $\mu g/m^3$. Fan *et al.* [17] proposed a deep LSTM to predict pollution in large geological regions. Wang *et al.* [15] combined a neural network and probabilistic analysis to predict large city pollutants in 2017. Krishan et al. [13] reported the use of LSTM for air pollution prediction in India in 2019. Jeya and Sankari [30] applied bidirectional LSTM to predict the air pollution in 2020. Some neural network variants (e.g. CNN-LSTM models [12], bidirectional LSTM [14], GRU [31]) were also applied. The advanced deep neural network focuses on improving overall accuracy. However, the challenge remains for improving the model interpretability and understanding the physical meanings of many high-dimensional features.

In recent years, there are many theoretical studies focusing on different aspects of interpretability of machine learning and deep neural network, for example, transparency, uncertainty, and explainability. The advances in these aspects have improved the ability of machine learning and deep neural network to explain and provide meanings in understandable ways to human. For deep neural networks, the most commonly used methods are post-explainability techniques [32], which include simplification, feature relevance analysis, visual explanation. Model-Agnostic Explanations (LIME) is one of the most know contributions to simplification approach [33]. It can approximate complex black box machine learning model with an interpretable model. This approach finds a balance between model fidelity and complexity. It reveals what is happening in the machine learning systems and what are the potential risks and how the systems can be trusted. Feature relevance analysis aims to provide understandable information to human, by ranking and measuring the influence and importance of each feature for the prediction. A popular feature relevance analysis method is SHapley Additive exPlanations (SHAP) [34], which explains the machine learning systems by calculating the contribution of each feature to the prediction instance. Visual explanation is another method to explain machine learning, but it is less common than the feature relevance analysis. The reason is that a visual explanation method is usually designed for a machine learning system with specific structures, and it is difficult to apply it to other systems [32]. For multi-layer deep convolutional and recurrent neural network, the feature relevance analysis has become a popular approach to improve interpretability. For example, Montavon *et al.* [35] developed a method which can decompose the classification decision into contributions of the input factors. Zeiler *et al.* [36] designed and applied a Deconvnet network to reveal parts of the image that produced the results. Another approach is to design a hybrid model structure which combines an interpretable model and a deep neural network model. For example, Krakovna [37] constructed a hybrid model which benefits form the interpretability of Hidden Markov Model (HMM) and the accuracy of recurrent neural network. Some other studies also employed fuzzy method to improve the model interpretability of machine learning models. Bougoudis *et al.* [23], for example, developed a fuzzy semi-supervised forecasting framework for air pollution prediction in Athens. The hybrid algorithm was developed based on Naïve Bayes classifier and fuzzy clustering, which can predict extreme air pollutants' values and explore the correlation between pollution and climatic conditions. While these approaches significantly improve the explainability and provide insights to human, their limitation in transparency still remains.

Traditional regression-based methods use intuitive and straightforward structures (e.g., polynomials). In 2013, Sampson *et al.* [27] employed a partial least square method to estimate $PM_{2.5}$ in USA. A regression model is proposed by Di *et al.* [26] to predict $PM_{2.5}$ in the Northeastern United States. In 2017, a similar method was applied to predict pollution in Hong Kong by Lee *et al.* [28]. Comparing to neural networks, traditional regression-based models are simple and interpretable, but not superior in prediction accuracy, especially for peak values prediction and high-dimension high-volume noisy data.

The NARMAX model is an advanced nonlinear regression-based model, which is developed for data modelling in the time, frequency, and spatiotemporal domains [2938]. It can automatically generate nonlinear features with physical meaning and pick out the most effective features to establish a linear-in-the-parameter model [3132]. The feature selection procedure is crucial in many real-world applications. In practice, the initial feature space is usually extremely large. If all the features are included in the model, it will be very time consuming to train the model. More importantly, the model can be overfitted and the most effective features cannot be identified, which cause performance drops in severe situations such as peak values. To address these concerns, the NARMAX model employs an efficient approach to generate nonlinear interpretable features by performing nonlinear conversations to time lagged linear features derived from raw time series input. Then, an Orthogonal Forward Regression (OFR) algorithm is applied to select the important features in a stepwise manner and an orthogonalization procedure is used to reduce the search space in each step. Benefit from this architecture, it usually takes much less time to identify sufficient features for prediction. Compared to traditional regression-based models, the NARMAX model not only considers the linear time lagged features, but also derives and identifies nonlinear time lagged features which can better capture the dynamics in complex nonlinear systems. In this way, the NARMAX significantly improve the prediction accuracy while holding the interpretable structure. The NARMAX model and its variants [29,38,39] have been successfully applied to a wide range of fields, including modelling of space weather [40,41], environment [42], EEG [43], etc.

In summary, various techniques have been developed on different aspects of model interpretability, for example, explainability, transparency. However, limited studies, so far, have focused on developing a single framework to cover all three aspects, transparency, explainability and peak value performance. In this work, we aim to develop an interpretable machine learning model that fills the existing gaps, by providing the following properties with one single framework: 1) the model uses a transparent structure that is understandable by human; 2) the model explores the importance of features and reveals the correlation between output and input; 3) the model improves the peak value prediction performance.

3. The proposed HIP-ML model

The basic idea of the proposed HIP-ML model is to design a hybrid model that combines an interpretable model and a deep neural network. Comparing to the Model-Agnostic Explanations and feature relevance analysis approaches, the hybrid structure has two main advantages. First, the hybrid model does not use any approximation to the deep neural network. The model itself is straightforward to use and interpretable. Second, the feature importance analysis can be integrated to the model structure selection procedure, so that it not only reveals the feature relevance but also provides a transparent model. Thus, the hybrid model can achieve both transparency and explainability with one single model framework.

The estimation of the HIP-ML model contains several steps. The first step is to generate candidate features (linear and nonlinear) from basic predictors (Section 3.1). The second step is to select discriminative features among these candidates and use them to build a 'stage-I' interpretable polynomial model (Section 3.2). The third step is to build a 'stage-II' neural network to fit the residual of 'stage-I' model, to further improve the prediction accuracy, especially for peak values (Section 3.3). The final HIP-ML model is constructed with stage-I and stage-II model (Section 3.3). A detailed diagram of the proposed architecture is presented in Fig. 1 and a diagram of integration the two stage models is presented in Fig. 2.

3.1. Candidate feature generation

Given an input–output process, the predictive model is built to predict output (response) variable $\boldsymbol{y}$ from multiple time series input (explanatory) variables$\boldsymbol{X}$. The input and output variables are defined as:

$$\boldsymbol{y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} \tag{1}$$

$$\boldsymbol{x_i} = \begin{bmatrix} x_i(1) \\ x_i(2) \\ \vdots \\ x_i(N) \end{bmatrix}, i = 1, 2, \cdots, R \tag{2}$$

$$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x_R}] \tag{3}$$

whereN, R are the number of samples and inputs, respectively. Assume that the aim is to predict $y(t)$ for D hours ahead, the predictors can be defined as:

$$y(t-D), \cdots, y(t-D-d_y), x_1(t-D)\cdots, x_1(t-D-d_x),$$
$$x_2(t-D), \cdots, x_2(t-D-d_x)\cdots, x_R(t-D), \cdots, x_R(t-D-d_x) \tag{4}$$

where D is the time delay; d_x and d_y indicates the maximum time-lags for input and output variables.

These predictors are considered as 'linear features' in linear regression-based models. However, they are usually insufficient to describe the complex nonlinear dynamics of the input–output process. For many real data problems, nonlinear features are more effective in predictive models [29]. Therefore, some extra nonlinear candidate features are derived from the linear candidate features.

First, we present the definition of the linear candidate features as $\boldsymbol{\varphi}^{linear} = \{\boldsymbol{x_r}, \boldsymbol{y_r}\}$, as follows [29]:

$$\boldsymbol{x}_r = [x_i(t-D), x_i(t-D-1), x_i(t-D-2), \cdots, x_i(t-D-d_x)] \tag{5}$$

$$\boldsymbol{y}_r = [y(t-D), y(t-D-1), y(t-D-2), \cdots, y(t-D-d_y)] \tag{6}$$

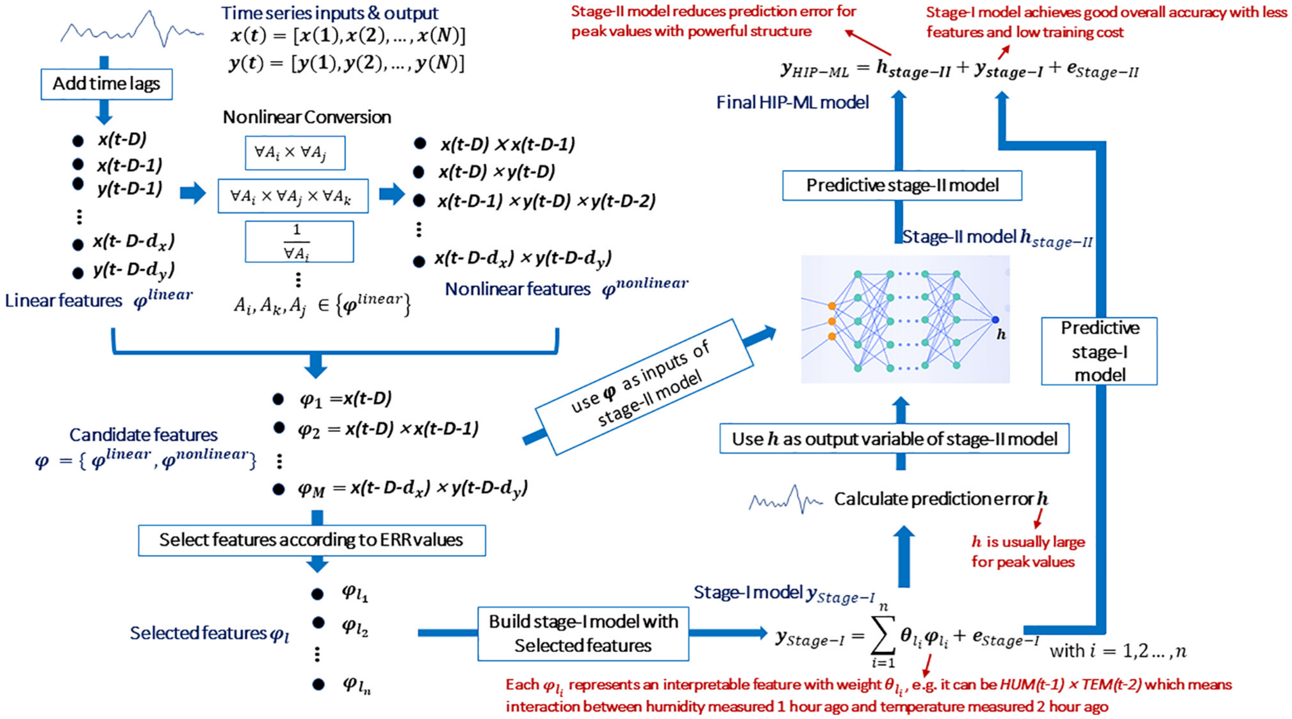
Time series inputs & output
$x(t) = [x(1), x(2), \ldots, x(N)]$
$y(t) = [y(1), y(2), \ldots, y(N)]$
Add time lags
x(t-D)
x(t-D-1)
y(t-D-1)
x(t- D-d_x)
y(t- D-d_y)
Linear features φ^{linear}
Nonlinear Conversion
$\forall A_i \times \forall A_j$
$\forall A_i \times \forall A_j \times \forall A_k$
$\frac{1}{\forall A_i}$
$A_i, A_k, A_j \in \{\varphi^{linear}\}$
x(t-D) × x(t-D-1)
x(t-D) × y(t-D)
x(t-D-1) × y(t-D) × y(t-D-2)
x(t- D-d_x) × y(t-D-d_y)
Nonlinear features $\varphi^{nonlinear}$
Candidate features
$\varphi = \{\varphi^{linear}, \varphi^{nonlinear}\}$
φ_1 = x(t-D)
φ_2 = x(t-D) × x(t-D-1)
φ_M = x(t- D-d_x) × y(t-D-d_y)
Select features according to ERR values
Selected features φ_l
φ_{l_1}
φ_{l_2}
φ_{l_n}
Build stage-I model with Selected features
Stage-I model $y_{Stage-I}$
$y_{Stage-I} = \sum_{i=1}^{n} \theta_{l_i}\varphi_{l_i} + e_{Stage-I}$ with $i = 1,2\ldots,n$
Each φ_{l_i} represents an interpretable feature with weight θ_{l_i}, e.g. it can be HUM(t-1) × TEM(t-2) which means interaction between humidity measured 1 hour ago and temperature measured 2 hour ago
use φ as inputs of stage-II model
Stage-II model reduces prediction error for peak values with powerful structure
Stage-I model achieves good overall accuracy with less features and low training cost
$y_{HIP-ML} = h_{stage-II} + y_{stage-I} + e_{Stage-II}$
Final HIP-ML model
Predictive stage-II model
Stage-II model $h_{stage-II}$
h
Predictive stage-I model
Use h as output variable of stage-II model
Calculate prediction error h
h is usually large for peak values

Fig. 1. Architecture of HIP-ML model.

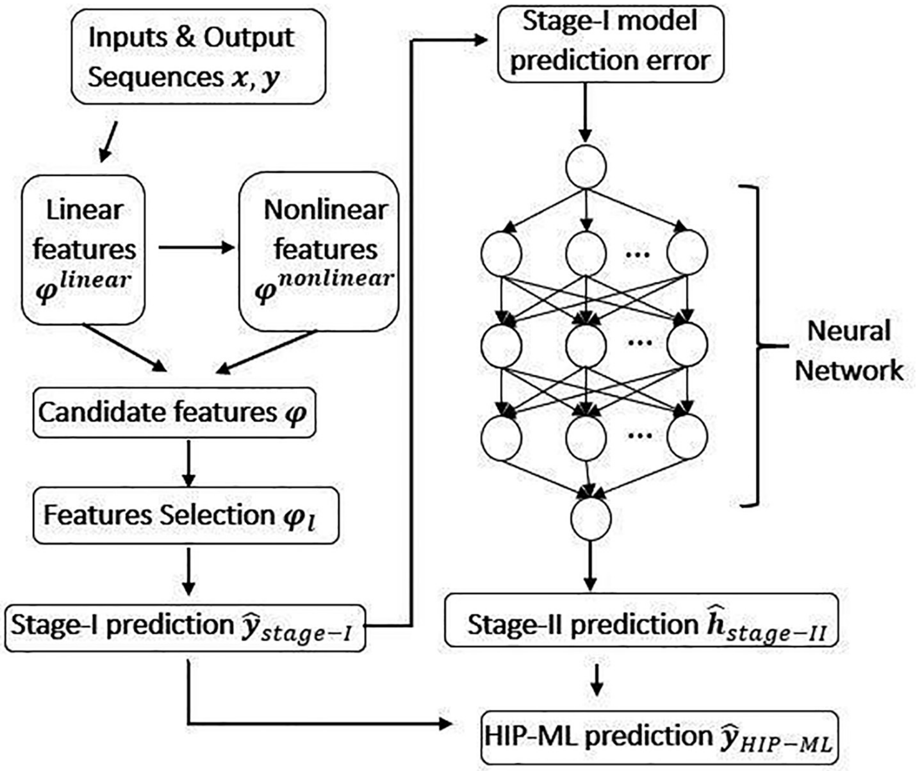
Inputs & Output Sequences x, y
Linear features φ^{linear}
Nonlinear features $\varphi^{nonlinear}$
Candidate features φ
Features Selection φ_l
Stage-I prediction $\hat{y}_{stage-I}$
Stage-I model prediction error
Neural Network
Stage-II prediction $\hat{h}_{stage-II}$
HIP-ML prediction $\hat{y}_{HIP-ML}$

Fig. 2. Integration of the two stage models.

where $\boldsymbol{x}_r$ are the historical records of R input variables with $i = 1, 2, \cdots, R$ and $\boldsymbol{y}_r$ are the historical records of output variable, also known as autoregressive variables.

Next, multiple nonlinear functions can be used to derive nonlinear features from φ^{linear}. To improve the model interpretability, the nonlinear features should be explainable. The HIP-ML model generates nonlinear candidate features by multiplying any of the linear candidate features at possible combinations. For example, a 2nd order nonlinear candidate feature can be presented as [29]:

$$\forall A_k \times \forall A_j \tag{7}$$

where $A_k, A_j \in \{\boldsymbol{\varphi}^{linear}\}$. In most situations, candidate features with a maximum 2nd or 3rd order can be sufficient to capture nonlinear combination nature from original inputs to achieve good nonlinear capability of the model [29]. In addition, a constant C is also considered as a candidate feature.

The final candidate features include linear candidate features, nonlinear candidate features and a constant, which are defined as:

$$\boldsymbol{\varphi} = \{\boldsymbol{\varphi}^{linear}, \boldsymbol{\varphi}^{nonlinear}, C\} = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M] \tag{8}$$

where M is the total number of the candidate features.

3.2. Feature selection and stage-I model estimation

Not all the candidate features $\boldsymbol{\varphi}_1 \cdots \boldsymbol{\varphi}_{\boldsymbol{M}}$ are useful for establishing the predictive model. If all the features are included in the model, the model will be extremely complex and hard to interpret. In addition, it will cause high computation cost and overfitting issue.

To reduce the feature space, a feature selection process is developed by incorporating the OFR algorithm [31,34] into the HIP-ML model, to select the most discriminative features among the candidates. The algorithm uses an Error Reduction Ratio (ERR) index as selection criterion, which measures the reduction of mean square error when adding a candidate feature to evaluate the model accuracy. The algorithm selects features in a forward stepwise manner. At each step, the algorithm first measures the ERR of each candidate features and select the one with highest ERR (Algorithm 1 step 8 $\sim$ 9); next, an orthogonalization process is applied to update the remaining candidate features (Algorithm 1 step 11); then, the algorithm calculates a model complexity determination criterion (Algorithm 1 step 12 $\sim$ 13), to decide whether stop the selection process, or move onto next step to continue the selection process.

Here, the orthogonalization procedure is applied to achieve the following objective: at each step, if the impacts of latter features overlap with those of the previous selected features, the overlapped part will not be measured again by ERR. In this way, the number of required features can be reduced. The model complexity determination criterion is adjustable prediction error sum of squares (APRESS) [44]. In general, the APRESS value decreases when a first few features are included in the model, because of the reduction of prediction error. When sufficient features are included, the penalty of complexity becomes significant and the APRESS value increases. The optimal number of features is at this turning point. Assume that n discriminative features are selected (Algorithm 1 step 16). we present them as $\boldsymbol{\varphi}_{\boldsymbol{l}} = \left\{\boldsymbol{\varphi}_{l_1}, \boldsymbol{\varphi}_{l_2}, \cdots, \boldsymbol{\varphi}_{l_n}\right\}$, in which $l_1 \cdots l_n$ are the indices. Note that n is usually much smaller than M (refer to experiment results in Section 4).

Then, the selected features are used to build the 'stage-I' model. The stage-I model utilizes an interpretable structure, similar to NARX [2938] model, as:

$$\boldsymbol{y}_{Stage-I} = f[\boldsymbol{\varphi}_{\boldsymbol{l}}] + \boldsymbol{e}_{Stage-I} \tag{9}$$

where $f[\cdot]$ represents the nonlinear function and $\boldsymbol{e}_{Stage-I}$ is the noise signal of the stage-I model. In many cases, the function $f[\cdot]$ uses a polynomial structure, so stage-I model can be written as:

$$\boldsymbol{y}_{Stage-I} = \sum_{i=1}^{n} \theta_{l_i} \boldsymbol{\varphi}_{l_i} + \boldsymbol{e}_{Stage-I} \tag{10}$$

where $\theta_{l_1} \cdots . \theta_{l_n}$ are the estimated weights of the selected features $\boldsymbol{\varphi}_{\boldsymbol{l_1}}, \boldsymbol{\varphi}_{\boldsymbol{l_2}}, \cdots, \boldsymbol{\varphi}_{\boldsymbol{l_n}}$ using least square estimator (Algorithm 1 step 17).

Comparing to other feature selection methods, the OFR algorithm has several unique advantages. First, the derived and selected features are fully interpretable. As defined in Section 3.1, the linear features are generated by adding time lags to original variables, and the nonlinear features are derived by performing nonlinear conversions to the linear features. For example, the nonlinear feature $y(t-1) \times x_2(t-2)$ indicates the interaction between the output variable at time $t-1$ and the second input variable at time $t-2$. The features clearly show what information is used to build the model and the associated weights of these features can be estimated. Second, the incorporated orthogonalization procedure can reduce the feature search space in each selection step. This will largely reduce the number of required features for building the model and the time cost for training the model. Although many other feature selection methods such as Principal Component Analysis (PCA) [21] and LSTM encoder [45] are very efficient in producing features that can improve prediction accuracy, the derived features are not fully transparent. Therefore, the OFR algorithm is more suitable for many real-world applications, where interpretability is highly desired to better explore the insights of the model and understand the driving factors. Taken air pollution prediction as an example, it is crucial to investigate the key factors that causes the pollution, to help decision makers better control the pollution status.

The stage-I model largely reduces the feature space and computational cost by effectively selecting the features. As the stage-I model uses a polynomial structure as fundamental for feature generation and to minimize the learning cost, it is therefore straightforward to explain the model with the meaning and weights of selected features, reflecting the importance of original input factors, impact of their combinations and system dynamics.

Algorithm 1 Feature selection and two-stage model estimation

```
1:  Input output vector $\boldsymbol{y}$, candidate features $\boldsymbol{\varphi}_i$ with
    $i = 1, 2, 3, \cdots, M$
2:  Initialize adjustable parameter $\alpha$
3:  Assign $\boldsymbol{r}_o \leftarrow \boldsymbol{y}$
4:  Assign $APRESS[0] \leftarrow 0$
5:  Assign $\boldsymbol{q}_i^{(1)} \leftarrow \boldsymbol{\varphi}_i$
6:  Initialize $s \leftarrow 1$
7:  while $APRESS[s] < APRESS[s-1]$ do
8:      Compute $ERR_i^{(s)} \leftarrow \frac{\left[y^T q_i^{(s)}\right]^2}{(y^T y)\left(q_i^{(s)^T} q_i^{(s)}\right)}$ for $i = 1, 2, \ldots M$
9:      Find $l_s \leftarrow arg \underset{1 \le j \le M, j \notin l}{max} \left\{ERR_j^{(s)}\right\}$
10:     Assign $\boldsymbol{q}_s \leftarrow \boldsymbol{q}_{l_s}^{(s)}$
11:     Compute $\boldsymbol{q}_i^{(s+1)} \leftarrow \boldsymbol{q}_i^{(s)} - \frac{\boldsymbol{q}_i^{(s)^T} \boldsymbol{q}_s}{\left(\boldsymbol{q}_i^{(s)^T} \boldsymbol{q}_s\right)} \boldsymbol{q}_s$ for $i = 1, 2, \cdots, M$
12:     Compute$\|\boldsymbol{r}_s\| \leftarrow \|\boldsymbol{r}_{s-1}\| - \frac{\left[\boldsymbol{r}_{s-1}^{\boldsymbol{T}} \boldsymbol{q}_s\right]^2}{\left(\boldsymbol{q}_s^{\boldsymbol{T}} \boldsymbol{q}_s\right)}$
13:     Compute $APRESS[s] \leftarrow \frac{1}{1 - \frac{s \times \alpha}{N}} \times \frac{\|\boldsymbol{r}_s\|}{s}$
14:     Updates $\leftarrow s + 1$
15: end while
16: Update selected features as $\boldsymbol{\varphi}_{\boldsymbol{l}} \leftarrow \left[\boldsymbol{\varphi}_{l_1}, \boldsymbol{\varphi}_{l_2}, \cdots, \boldsymbol{\varphi}_{l_n}\right]$
17: Compute weights $\boldsymbol{\theta}_l \leftarrow \left[\boldsymbol{\varphi}_{\boldsymbol{l}}^{\boldsymbol{T}} \boldsymbol{\varphi}_{\boldsymbol{l}}\right]^{-1} \boldsymbol{\varphi}_{\boldsymbol{l}}^{\boldsymbol{T}} \boldsymbol{y}$
18: Compute prediction error $\boldsymbol{h} \leftarrow \boldsymbol{y} - \sum_{i=1}^{n} \theta_{l_i} \boldsymbol{\varphi}_{l_i}$
19: Compute $\boldsymbol{h}_{\boldsymbol{r}} \leftarrow [h(t-D), h(t-D-1), \cdots, h(t-D-d_h)]$
20: Train the stage-II model $g[\boldsymbol{h}_{\boldsymbol{r}}, \boldsymbol{\varphi}]$ to fit $\boldsymbol{h}$
21: Compute stage-II model prediction $\widehat{\boldsymbol{h}}_{stage-II}$
22: Compute HIP-ML prediction
    $\widehat{\boldsymbol{y}}_{HIP-ML} \leftarrow \sum_{i=1}^{n} \theta_{l_i} \boldsymbol{\varphi}_{l_i} + \widehat{\boldsymbol{h}}_{stage-II}$
23: Output $\boldsymbol{\varphi}_{\boldsymbol{l}}, \boldsymbol{\theta}_l$ (stage-I model), $\boldsymbol{g}[\boldsymbol{h}_{\boldsymbol{r}}, \boldsymbol{\varphi}]$(stage-II model),
    $\widehat{\boldsymbol{y}}_{HIP-ML}$ (HIP-ML model prediction)
```

3.3. Stage-II model estimation and integrated deep neural network

The stage-I model can usually produce prediction with good overall accuracy. However, at some severe situations when sudden data peak appears (e.g., a significant change on $PM_{2.5}$), the stage-I model might not be sensitive enough to capture all the underlying dynamics. These sudden changes often present in real data and increase prediction error of peak periods.

To improve the accuracy for these peak values, an extra model structure is required to describe the complex dynamics and fit the prediction error of severe situations. The HIP-ML model uses such a 'stage-II' model to absorb the residual of stage-I model. The residual of stage-I model is calculated as follows (Algorithm 1 step 18).

$$\boldsymbol{h} = \boldsymbol{y}_{stage-I} - \hat{\boldsymbol{y}}_{stage-I} \tag{11}$$

The stage-II model employs a feedforward deep neural network with one input layer, one output layer and several fully connected layers. The numbers of the fully connected layers and neurons are determined by some trial experiments (details in Section 4.2). The scaled conjugate gradient backpropagation algorithm is applied to train the neural network. The output (response) variable of the stage-II model is the residual $\boldsymbol{h}$. The inputs of the stage-II model contain two parts. The first part is generated features in Section 3.1. The second part is the autoregressive variables of the residual $\boldsymbol{h}$, acting as a moving error average. They are defined as:

$$\boldsymbol{h_r} = [h(t-D), h(t-D-1), \cdots, h(t-D-d_h)] \tag{12}$$

where d_h is the maximum time-lag (Algorithm 1 step 19). The stage-II model is built as follows:

$$\boldsymbol{h}_{stage-II} = g[\boldsymbol{h_r}, \boldsymbol{\varphi}] + \boldsymbol{e}_{Stage-II} \tag{13}$$

where $g[\cdot]$ represents the neural network and $\boldsymbol{e}_{Stage-II}$ is the noise signal of stage-II model (Algorithm 1 step 20).

Based on stage-I and stage-II model, the hybrid HIP-ML model is defined as:

$$\boldsymbol{y}_{HIP-ML} = \boldsymbol{y}_{stage-I} + \boldsymbol{h}_{stage-II} + \boldsymbol{e}_{Stage-II} \tag{14}$$

The model (14) is used to generate the prediction (Algorithm 1 step 21 $\sim$ 22). The above feature selection and two-stage estimation procedures are summarized in Algorithm 1. The architecture of HIP-ML model is presented in Fig. 1. In the figure, each φ_{l_i} represents an interpretable feature with weight θ_{l_i}, e.g., it can be Hum (t-3) × Tem(t-5) which means interaction between humidity measured 1 h ago and temperature measured 2 h ago. Stage-I model can achieve good overall accuracy with less features and low training cost. The residual of stage-I model h is usually large for peak values. Stage-II model reduces prediction error for peak values with powerful structure. A diagram is presented in Fig. 2 to illustrate how the two sub-models are merged.

The hybrid HIP-ML model has several advantages. First, only small number of features are used to fit most of the input–output process. Therefore, the model can provide interpretable representation and possible overfitting issue are significantly eliminated. Second, benefit from the stage-II model, the peak values prediction is improved. Third, since the overall residual of stage-I model is usually small, the training time of complex deep neural network in stage-II model is reduced. This advantage is crucial when prediction is required for applications with high-dimensional data input and real-time performance.

3.4. Evaluation of prediction accuracy

To evaluate the prediction, the correlation coefficient (CC), prediction efficiency (PE), and normalized root mean square error (NRMSE) between the model predictions and true observations are calculated. They are defined as:

$$CC = Cov_{op}/(\sigma_{observed} * \sigma_{predicted}) \tag{15}$$

$$PE = 1 - \sigma^2_{error}/\sigma^2_{observed} \tag{16}$$

$$NRMSE = \sqrt{\frac{1}{N}\left(\hat{\boldsymbol{y}} - \boldsymbol{y}\right)^2} / [\max(\boldsymbol{y}) - min(\boldsymbol{y})] \tag{17}$$

where $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are the observations and predictions; $\sigma_{observed}$, $\sigma_{predicted}$ and σ_{error} are the standard deviations of observations, predictions and prediction error; Cov is the covariance between observations and predictions.

4. Results and discussions

This section presents the data description, data pre-processing, experiment, results, and discussions. First, some pre-processing methods were developed on Python to read the raw data file, extract the variables, and remove the missing values and outliers. Second, the proposed model was constructed on Matlab to predict PM2.5 for multiple prediction time scales. The deep neural network of stage-II model was implemented using Matlab deep learning toolbox. In addition, several state-of-the-art deep neural networks (e.g., LSTM, GRU) were built using Python Keras for comparison purpose. Finally, the results were evaluated by several prediction accuracy criteria and insights of the results were discussed.

4.1. Data and pre-processing

A representative air pollution dataset from UC Irvine Machine Learning Repository [46] is used to evaluate the proposed HIP-ML model. The dataset includes air pollution, season and weather measurements (e.g. $PM_{2.5}$, air pressure, humidity, temperature, season, wind direction, etc.) from 2012 to 2015 in Beijing, China. Descriptions of the measured variables are presented in Table 1. The variables in Table 1 were extracted from the raw dataset. To derive effective features for building the model, pre-processing methods need to be applied to clean the data, and then the processed data can be used to derive time lagged linear and nonlinear interpretable features following the procedures in Section 3.1.

Due to the sensor failure, measurement error, and data transmission error, the raw dataset contains a lot of missing values and outliers. Some pre-processing methods were applied to

Table 1
Data Descriptions.

Variable	Name	Description
$\boldsymbol{y}$	PM	$PM_{2.5}$ concentration [ug/m^3]
$\boldsymbol{x}_1$	Sun	Sunlight [Yes/No]
$\boldsymbol{x}_2$	Dew	Dew Point [Celsius Degree]
$\boldsymbol{x}_3$	Hum	Humidity [%]
$\boldsymbol{x}_4$	Pre	Pressure [hPa]
$\boldsymbol{x}_5$	Tem	Temperature [Celsius Degree]
$\boldsymbol{x}_6$	CWS	Cumulated wind speed [m/s]
$\boldsymbol{x}_7$	HPre	Hourly Precipitation [mm]
$\boldsymbol{x}_8$	CPre	Cumulated Precipitation [mm]
$\boldsymbol{x}_9$	Spr	Spring [Yes/No]
$\boldsymbol{x}_{10}$	Sum	Summer [Yes/No]
$\boldsymbol{x}_{11}$	Aut	Autumn [Yes/No]
$\boldsymbol{x}_{12}$	Win	Winter [Yes/No]
$\boldsymbol{x}_{13}$	NE	North East Wind [Yes/No]
$\boldsymbol{x}_{14}$	NW	North West Wind [Yes/No]
$\boldsymbol{x}_{15}$	SE	South East Wind [Yes/No]
$\boldsymbol{x}_{16}$	CV	No Wind [Yes/No]
$\boldsymbol{x}_{17}$	SW	South West Wind [Yes/No]

address these issues. The missing values were filled using interpolation methods. The outliers are data points that differ significantly from surrounding observations, which were analyzed and removed using smoothing and thresholding methods. For categorical variables, such as seasons and wind directions, a dummy variable was generated to represent each class in a category. The dummy variable only takes a value of 0 or 1 to indicate the absence or presence of a categorical class. With dummy variables, each class of a categorical variable is represented equally, so that the contributions of these classes can be measured fairly.

The time series of pre-processed data of year 2015 are presented in Fig. 3. The data contains peak $PM_{2.5}$ values measurements, indicating high pollution level. When $PM > 400$, the pollutants are extremely harmful and may cause significant health problems. Large variations, uncertainty, and rapid changes in this dataset are challenging to predictive models. There are in total around 26,000 data points captured at every hour from 2012 to 2015. We used 75% of the data for training, and the remaining 25% data for testing.

4.2. The identified HIP-ML models

Three HIP-ML models are built to predict PM2.5 for 1, 3 and 6 h ahead. They are named as'1-hour HIP-ML model', '3-hour HIP-ML model' and '6-hour HIP-ML model', respectively. The inputs are the 17 variables $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{17}]$ and the output $\boldsymbol{y}$ is the PM (Table 1). The maximum time-lags d_x, d_y are determined by some trial experiments. Based on our experiments, the maximum degree of the nonlinear features is chosen to be 2. As described in Section 3.1 and 3.2, the stage-I models are firstly built with a number of selected features. The selected features and associated weights, ERR values, statistical t-test values of 1,3,6-hour HIP-ML models

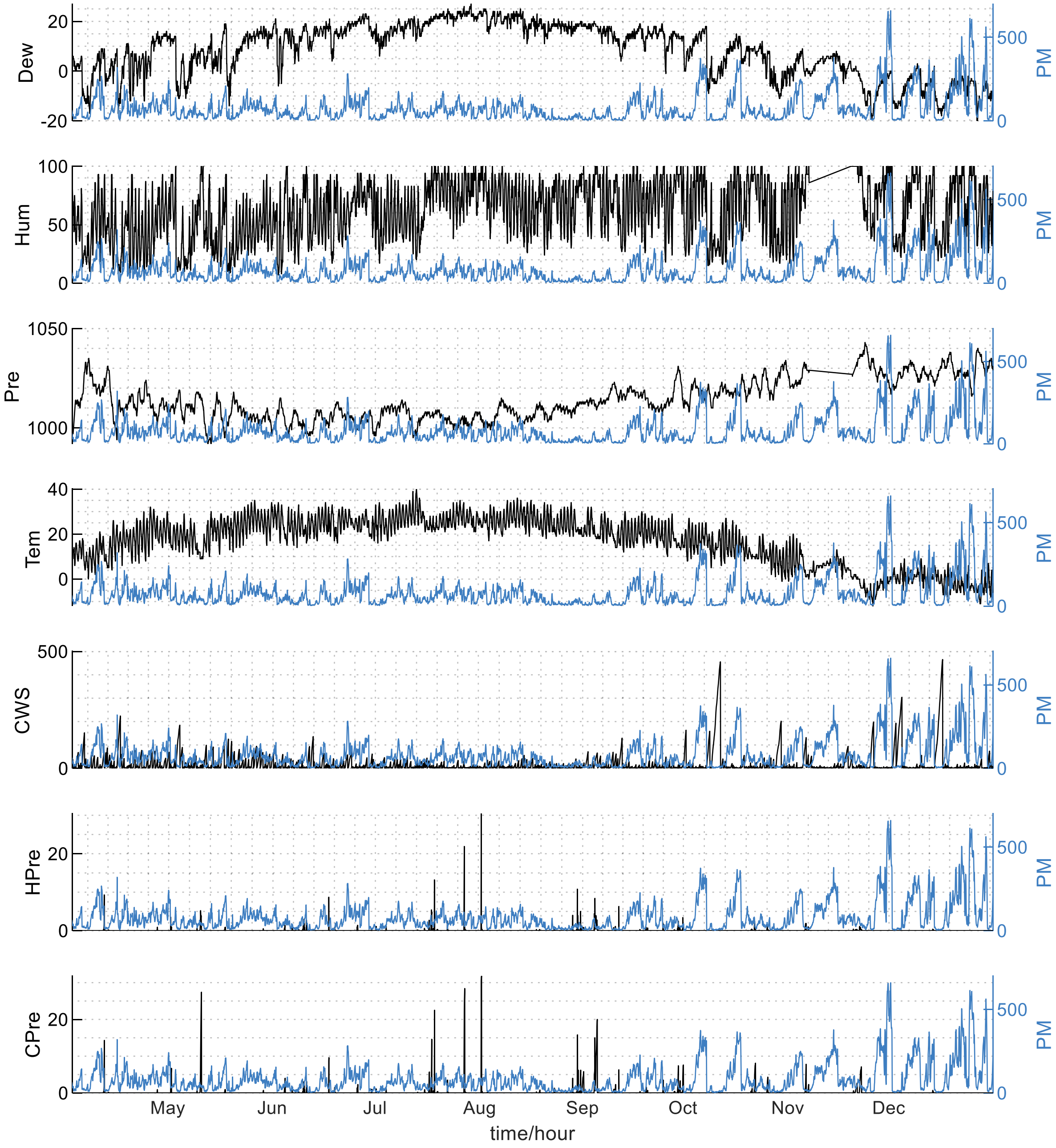

Fig. 3. Time series of $PM_{2.5}$ and external variables of year 2015.

are presented in Table 3. Note that these features are interpretable and provide information of which of the variables and measurements are used to build the model. The time lags also indicate the time dependencies between the prediction and explanatory variables. For example, PM(t-1) means the historical PM2.5 values 1 h ago, Hum(t-3) × Pre(t-5) means the interactions between humidity 3 h ago and precipitation 5 h ago.

The number of initial candidate features is 1378 and the models are constructed with only 7 $\sim$ 8 of these features. These features are selected according to their ERR values, which measure their impacts to $PM_{2.5}$ (details in Section 3.2 and Algorithm 1). Due to the orthogonalization process, the ERR of top-ranked selected features (e.g., the first selected feature) are usually larger than those of the latter features. We present the ERR values in Fig. 4 to show the variations among the selected features.

The statistical hypothesis *t*-test is applied to validate the significance of the selected features [39]. From the *t*-test results in Fig. 4, the *t*-test values of all the selected features are larger the 95% confidence value ($\sim$0.95), indicating that the selected features and estimated weights are significant.

Then, the stage-II models are estimated to fit the residuals of stage-I models, as described in Section 3.3. From some trial experiments, the number of layers and neurons of the neural networks are determined as follows: the 1-hour stage-II model contains 1 fully connected layer with 10 neurons; the 3-hour stage-II model contains 2 fully connected layers with 20 neurons; the 6-hour stage-II model contains 6 fully connected layers with 10 neurons. We optimize the neural network structure for each prediction time scale, to better describe the relationships between the output and input variables. As the prediction time scale increases, the time delay between output and inputs is increased, which propagates the uncertainty and makes the nonlinear dynamics more complex. For example, the 1-hour-ahead prediction relies on the input factors measured 1–3 h ago with minimum time lag of 1; the 3-hour-ahead prediction relies on the input factors measured 3–5 h ago with the minimum time lag of 3. Comparing to the 1-hour model, more layers and neurons can help improve the prediction accuracy. Because the stage-I models can explain most of the variations in PM2.5, stage-II models do not use very complex structure to fit the residual. Thus, the training time cost of the stage-II models is much lower than individual neural networks.

The final HIP-ML models are built based on the stage-I models and the stage-II models. For example, the prediction of 6-hour HIP-ML model is calculated as:

$$\begin{aligned}\widehat{PM}\ (t) = {} & 0.0482 \times \mathrm{Pre}(t-8) \times PM(t-6) + 5.8910 \\ & \times Sun(t-6) \times Sun(t-8) - 0.0004 \times PM(t-7) \\ & \times PM(t-7) + 0.1359 \times NW(t-6) \times PM(t-7) \\ & + 31.5028 \times Sun(t-6) \times Win(t-6) - 0.0473 \\ & \times \mathrm{Pre}(t-6) \times PM(t-6) + 0.0137 \times \mathrm{Pre}(t-6) \\ & \times SE(t-6) + 19.0961 \times Sun(t-6) \times Aut(t-6) \\ & + \hat{h}_{stage-II}(t) \end{aligned} \tag{18}$$

where $\hat{h}_{stage-II}(t)$ represents prediction of the stage-II model.

4.3. Overall performance and peak value prediction accuracy

We evaluate the prediction accuracy of the HIP-ML models by several criteria (Section 3.4). For comparison purpose, the results are compared with some other machine learning and deep neural network models, including feedforward neural networks, Lasso models, LSTMs and GRUs. The feedforward neural networks contain one input layer, several fully connected layers and one output layer. The LSTMs and GRUs were optimised based on literature [45]. The LSTMs are composed of one input layer, one output layer, several LSTM layers and Relu layer. The GRUs contain one input layer, one output layer, several GRU layers and Relu layer. We also compare our results with some previous air pollution studies from literature.

Overall prediction accuracies of these models are presented in Table 3. The predictions were evaluated by CC, PE and NRMSE between the predicted PM2.5 and observed PM2.5 in test datasets. High CC and PE values indicate the better fit, and small NRMSE values indicate the lower error. Whereas CC explains the correlation between the prediction and observation, PE (also known as R-Squared) statistically measures how much variation can be explained and NRMSE measures the unexplained variation. Note that it is normal that the accuracy of the 6-hour model is lower than that of 1-hour and 3-hour model, because the model uncertainty becomes stronger when increasing the prediction time scale.

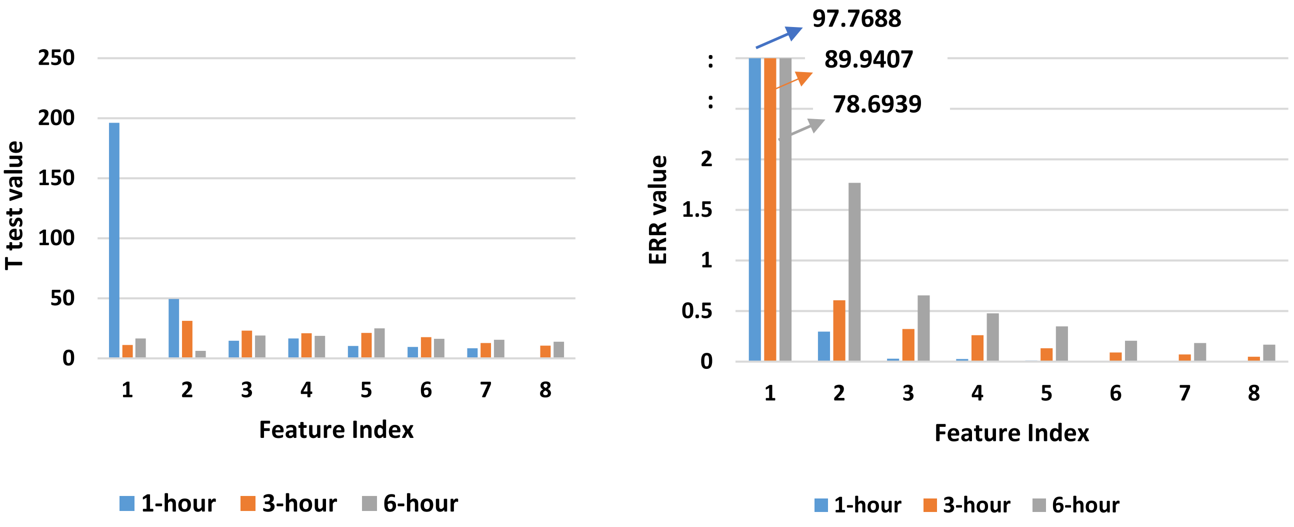

Fig. 4. The statistical *t*-test values (left figure) and ERR values (right figure) and of selected features of 1,3,6-hour models.

From the results, the HIP-ML models outperform the neural networks, Lasso models, LSTMs and GRUs with respect to overall prediction accuracy. Lasso models have the worst performances with low 1-hour-ahead prediction accuracy, and they perform poorly for 3 and 6-hour-ahead prediction. This is becasue Lasso models do not generate nonlinear features and use only linear features to build the predictive polynomial model. For this complex nonlinear system, the simple structures and features cannot effectively capture the dynamics. The proposed HIP-ML models, neural networks, LSTMs and GRUs achieved similar accuracies for 1-hour ahead prediction. Comparing to 3,6-hour ahead prediction, the time dependency of 1-hour-ahead prediction is the simplest due to the small time gap so all these networks work well. When generating 3,6-hour-ahead prediction, the HIP-ML models shows better prediction capacity than neural networks, LSTMs and GRUs with regards to PEs and NRMSEs. This indicates that the HIP-ML models better explain the variation in the $PM_{2.5}$.

The comparison of prediction and observation of 1,3,6-hour HIP-ML models are presented in Fig. 5. From the figure, the predictions of HIP-ML models have very small errors for periods when $PM_{2.5}$ is at low level. There are some accuracy drops during peak periods when $PM_{2.5}$ is larger than 400. It is normal for many time series prediction problems, as the models are usually trained on imbalanced data where the number of normal period samples is significantly higher than the number of the peak samples. As a result, the model usually fits the dynamics of normal period well but may fail to capture some characteristics in peak periods. The key for improving prediction for peak values lies on identifying the most important features that represents the system dynamics of both normal and peak periods. To evaluate the model performances of peak periods, RMSEs of the model predictions are calculated for the period when $PM_{2.5}$ is larger than 400. The RMSEs of peak value predictions of HIP-ML models, neural networks, LSTMs and GRUs are presented in Fig. 6. From the results, the peak value prediction errors of HIP-ML models are significantly lower than those of the neural networks, LSTMs and GRUs. For 1-hour-ahead peak value prediction, the RMSE of HIP-ML model is 36.59, around 43% lower than those of the neural network and LSTM. For 3-hour-ahead peak value prediction, neural network and LSTM have 17% and 26% larger errors than HIP-ML model. For 6-hour-ahead peak value prediction, all the models have significant performance drops and the RMSEs are larger than 200. Under this strong uncertainty, the HIP-ML model outperforms neural network and LSTM by 5% and 20%, respectively. Fig. 7 presents the scatter plots of the models, which provides more detailed comparison between prediction and observation. Clearly, all the models have good performances when $PM_{2.5}$ is lower than 400. Some points of HIP-ML models are slightly closer to the 45-degree reference line, showing that they have slight advantages at some time points. For the prediction when $PM_{2.5}$ is larger than 400, the scatter points of neural networks, LSTMs and GRUs are far away from the 45-degree reference line, indicating significant large errors. It can be also found that the predicted values are usually lower than the observed values. In these peak periods, the scatter points of HIP-ML model predictions are clearly closer to the 45-degree reference line, indicating that the prediction is significantly more accurate. These results indicate

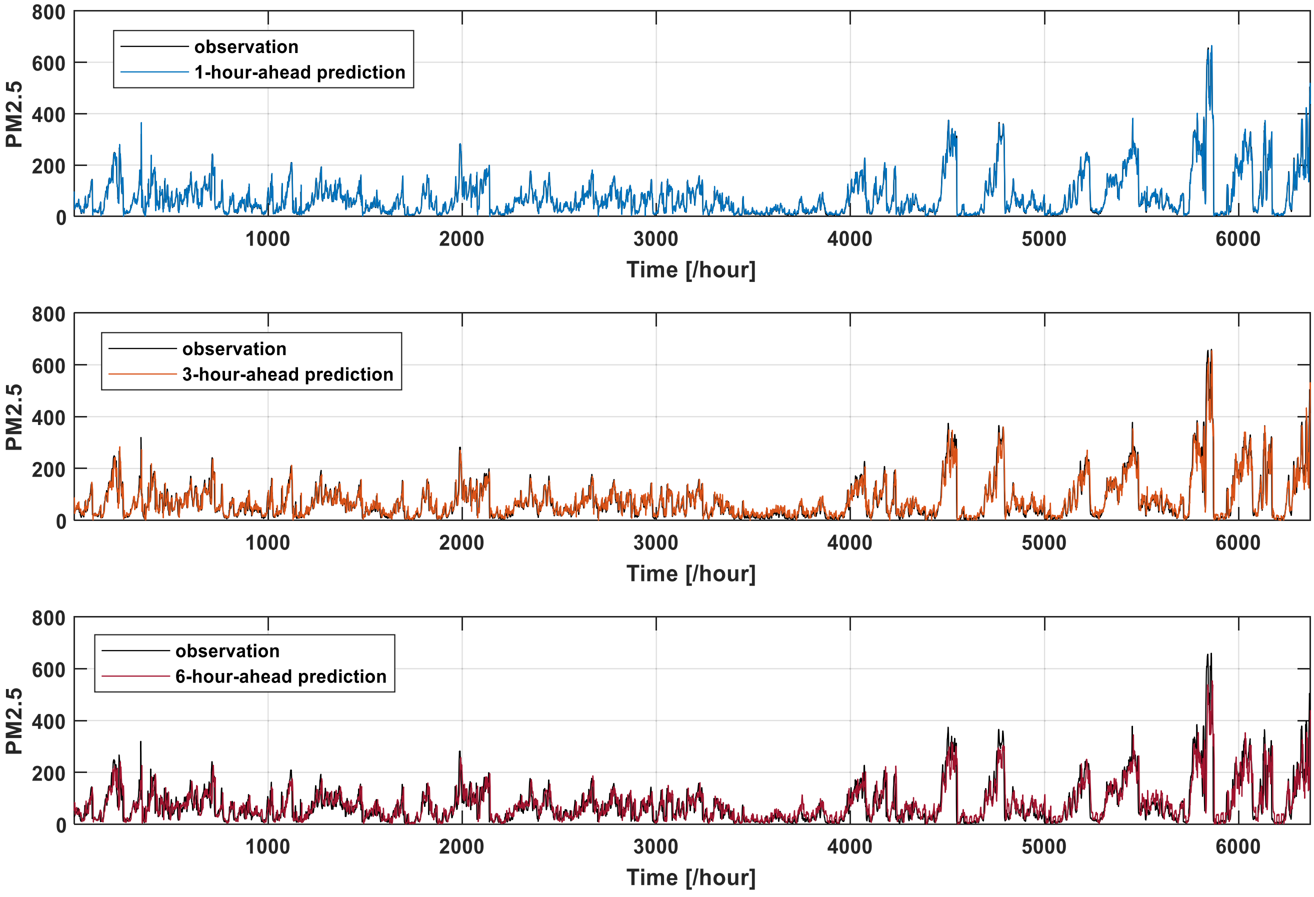

Fig. 5. Observed and predicted $PM_{2.5}$ of 1, 3, 6-hour HIP-ML models on test dataset.

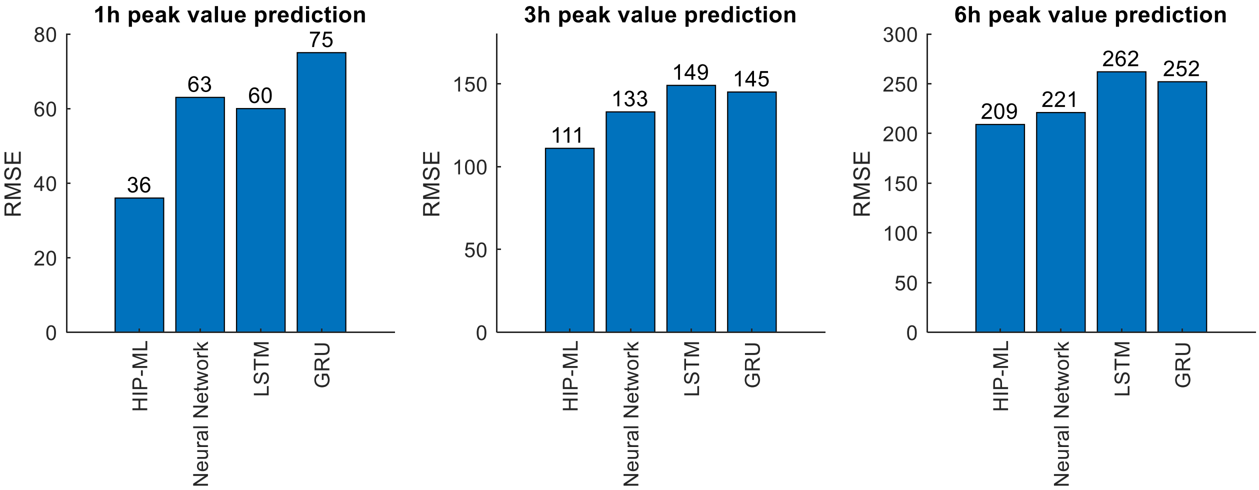

Fig. 6. RMSEs of peak value predictions ($PM_{2.5}$ greater than 400) of HIP-ML models, neural networks, GRUs and LSTMs.

that the HIP-ML model can generate better $PM_{2.5}$ prediction for severe situations when $PM_{2.5}$ is at a high level.

We also compare our results with some studies on air pollution prediction of Beijing from literature. The prediction efficiency of our 6-hour HIP-ML model is 64.81%, which is higher than the accuracy of 60% reported in [20]. The correlation coefficient and RMSE of our 1-hour HIP-ML model are 0.9855 and 15.0835. The results are better than those in [22], which reported correlation coefficient and RMSE of 0.90 and 23.69, respectively. Note that there might exist some bias when using these results to compare the model, as different datasets were used in these studies.

Overall, the HIP-ML models have better overall and peak value prediction performances than other models. The improvement of RMSEs appears to be very significant for peak value prediction, indicating that the architecture of the HIP-ML model works well on this air pollution prediction problem. The feature generation, feature selection and hybrid model structure can capture the complex nonlinear dynamics of the target system. More importantly, the interpretability of the model is significantly improved, as the identified features (Table 2) are explainable. Detailed discussions about model interpretability will be presented in next section.

4.4. Model interpretability

From Table 2, different combinations of interpretable features were used to build 1,3,6-hour predictive models. For example, linear features such as PM(t-1), PM(t-2) and nonlinear features such as NW(t-1) × PM(t-2), Pre(t-3) × Pre(t-3), Sun(t-1) × Win(t-1), HPre(t-1) × PM(t-1), Pre(t-1) × NW(t-3) are identified as important features for 1-hour-ahead prediction. According to the ERR values, the linear features have higher impact than nonlinear features, indicating that the 1-hour-ahead prediction highly depends on the historical PM2.5 values. This might be because it will it may take a longer time for weather conditions to affect PM2.5. As for 3-hour predictive models, all the identified features are nonlinear, indicating there are complex nonlinear dynamics. The weather factors such as precipitation, humidity and wind directions are selected as top-ranked nonlinear features, for example, Pre(t-5) × PM(t-3), Hum(t-5) × PM(t-4), NW(t-3) × PM(t-4), Hum(t-3) × Pre(t-5). The historical PM2.5 variables are used in these interaction terms. These show that the external weather factors play key roles in 3-hour predictive model and the effects of autoregressive variables (historical PM2.5) are weaker than those in 1-hour model. In this case, some weather factors (e.g., rain, wind) effect PM2.5 significantly. Similar to 3-hour model, some nonlinear features are used in the 6-hour model. It is reasonable as the complexity of the nonlinear dynamics increases when the prediction time gap goes up. In this situation, the impact of sunlight increased, as Sun(t-6) × Sun(t-8) was ranked as the second important feature. Precipitation, humidity and historical PM2.5 were still found to be key impact factors.

Based on the above observations, the factors (e.g., weather, season) in these features can significant affect $PM_{2.5}$. To learn the impact of various complex factors, we summarize the most frequently selected inputs (explanatory) variables in the models. First, the autoregressive variables PM(t-1), PM(t-2), PM(t-7) are always used in 1-hour, 3-hour and 6-hour models, revealing that the $PM_{2.5}$ prediction depends on its historical measurements. Then, the input variables ‘Pressure’, ‘North West’, ‘Winter’, ‘Sunlight’ are frequently used in the 1-hour, 3-hour and 6-hour models, which indicates that these factors have significant impacts on 1, 3 and 6-hour-ahead $PM_{2.5}$ predictions. Next, we also notice that the variable ‘Hourly Precipitation’ is only used in 1-hour model; the variable ‘Humidity’ is only used in the 3-hour model; the variable ‘Southeast Wind’ is only used in 6-hour model.

These results provide us the insight that the following factors significantly affect the 1, 3, 6-hour ahead $PM_{2.5}$: historical $PM_{2.5}$, pressure, northwest wind, winter, sunlight, hourly precipitation, humidity, southeast wind. With regard to the role of humidity, our finding re-confirms the conclusion of previous studies [7] that anti correlation between relative humidity and PM2.5 is associated to the occasional penetration of coastal air. One study [10] found that the rain can decrease air pollution, which partially supports our findings that the opposite situation (sunlight) can be correlated to air pollution. As for the importance of wind and season, our finding is consistent with that reported in [12] in two aspects. First, Beijing’s wind facilitates air diffusion and increases the proportion of clean air. Second, winter heating during winter autumn causes high proportion of serious pollution. Our results are in line with other previous research outcomes [13] which claimed the importance of meteorological factors. More importantly, our results contribute to the literature by identifying the interaction terms, which are ignored in the prior studies, rather than a single factor that as determinants of air pollution. For example, the interaction variable of sunlight and winter can increase the air pollution, while the interaction between precipitation and northwest wind can decrease the air pollution.

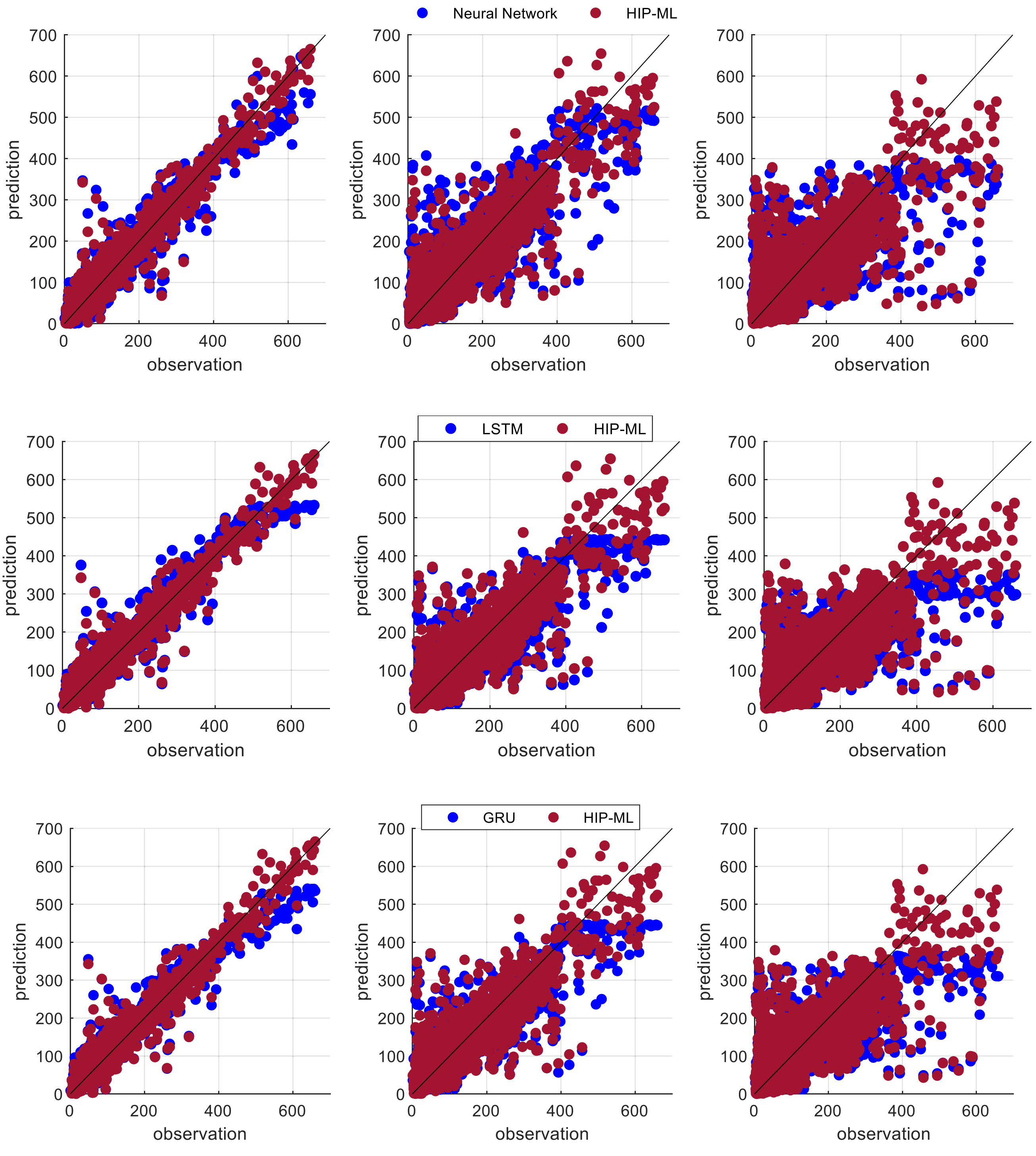

Fig. 7. Scatter plots of 1-hour (left), 3-hour (middle), 6-hour (right) HIP-ML models, neural networks, LSTMs and GRUs.

5. Conclusion

This study proposes a novel interpretable and hybrid HIP-ML model framework for time series-based air pollution prediction. The HIP-ML model is superior in model interpretability over other machine learning methods and could better explain the input–output behaviour or cause-effect relationships of a predictive system. Specifically, the proposed model could provide cost-effective and transparent representations to reveal the contributions of the selected features. Furthermore, our model could identify critical interaction features, combining influencing factor with its time lag, in $PM_{2.5}$ forecasting. In addition, the HIP-ML model could also improve the peak value predictions accuracy through the stage-II model.

In this article, we apply the HIP-ML model to predict $PM_{2.5}$ for 1, 3, and 6 h ahead. The experimental results show that historical $PM_{2.5}$, pressure, north west wind, winter, sunlight, and their interactions have significantly influence on PM2.5 forecasting. Our experimental results show that the overall prediction accuracy (measured by correlation coefficient) of 1, 3 and 6-hour-ahead pre-

Table 2
Selected features for 1-hour, 3-hour and 6-hour HIP-ML models.

Model	No.	Selected Features	Estimated Weights	*t*-test values	ERR (100%)
1-hour HIP-ML model	1	PM(t-1)	1.3051	196.2484	97.7688
	2	PM(t-2)	−0.3335	49.3611	0.2970
	3	NW(t-1) × PM(t-2)	−0.0358	14.5616	0.0294
	4	Pre(t-3) × Pre(t-3)	0.000004	16.5116	0.0263
	5	Sun(t-1) × Win(t-1)	3.8374	10.4140	0.0087
	6	HPre(t-1) × PM(t-1)	−0.0307	9.3599	0.0078
	7	Pre(t-1) × NW(t-3)	−0.0025	8.3911	0.0065
3-hour HIP-ML model	1	Pre(t-5) × PM(t-3)	0.0228	11.0474	89.9407
	2	Hum(t-5) × PM(t-4)	−0.0047	31.3410	0.6072
	3	NW(t-3) × PM(t-4)	−0.1167	22.9613	0.3240
	4	Hum(t-3) × Pre(t-5)	0.0002	20.9191	0.2624
	5	Sun(t-3) × Win(t-3)	16.8367	21.2385	0.1334
	6	Hum(t-3) × PM(t-3)	0.0031	17.6201	0.0908
	7	Sun(t-3) × Aut(t-3)	11.0431	12.6282	0.0707
	8	Pre(t-3) × PM(t-3)	−0.0218	10.5668	0.0476
6-hour HIP-ML model	1	Pre(t-8) × PM(t-6)	0.0482	16.4944	78.6939
	2	Sun(t-6) × Sun(t-8)	5.8910	6.3500	1.7684
	3	PM(t-7) × PM(t-7)	−0.0004	19.1062	0.6535
	4	NW(t-6) × PM(t-7)	−0.1359	18.8329	0.4786
	5	Sun(t-6) × Win(t-6)	31.5028	25.0312	0.3480
	6	Pre(t-6) × PM(t-6)	−0.0473	16.1792	0.2078
	7	Pre(t-6) × SE(t-6)	0.0137	15.5856	0.1831
	8	Sun(t-6) × Aut(t-6)	19.0961	13.8496	0.1679

Table 3
Prediction accuracies of 1, 3, 6-hour-ahead predictions (CC: Correlation Coefficient, PE: Prediction Efficiency, N-RMSE: Normalized RMSE).

Model	CC	PE	N-RMSE
1-hour neural network	0.9855	97.13%	0.0228
1-hour LSTM	0.9804	96.12%	0.0268
1-hour GRU	0.9789	96.77%	0.0285
1-hour Lasso model	0.6584	38.71%	0.1128
1-hour HIP-ML model	0.9855	97.12%	0.0229
3-hour neural network	0.9137	83.43%	0.0550
3-hour LSTM	0.9148	82.46%	0.0570
3-hour GRU	0.9164	83.43%	0.0552
3-hour Lasso model	0.0000	00.00%	0.1454
3-hour HIP-ML model	0.9253	85.58%	0.0511
6-hour neural network	0.8052	64.81%	0.0799
6-hour LSTM	0.8157	65.34%	0.0799
6-hour GRU	0.8147	65.83%	0.0795
6-hour Lasso model	0.0000	00.00%	0.1456
6-hour HIP-ML model	0.8294	68.78%	0.0752

diction is 0.9855, 0.9253 and 0.8294, respectively, indicating the state-of-the-art prediction performance achieved by the HIP-ML models. Meanwhile, the proposed model could reduce the training time and retain prediction accuracy through a small set of explainable combined features. This merit is crucial for 'big' data analysis with large high-dimensional datasets.

The interpretability of machine learning models is highly desired in many real-world applications. Concerning air pollution monitoring, our model is able to analyse and identify the sources of $PM_{2.5}$, and aid public organisations and government in developing efficient pollution mitigation policies. Although we utilize Beijing as an example for PM2.5 forecasting in this article, the proposed model can be applied to other locations as well as applications. Further, the proposed architecture employs general model structures for time series prediction, so it can also be extended and applied to a wide range of application scenarios. Examples of these applications include space weather [40,41], environment [42], EEG [43], social science [47], chemical [48], etc. The HIP-ML model can enhance the prediction performances and bring the predictive models to next level, by improving the model interpretability and peak values predictions.

For future work, we will further explore the effectiveness of the proposed architecture for time series prediction. We will investigate the performance of other deep neural networks and apply the HIP-ML model to other application scenarios. In addition, uncertainty analysis has become crucially important in recent years, as the quantification of model uncertainty brings insights of model interpretability and robustness. We will explore the solutions of integrating the proposed HIP-ML model and uncertainty analysis methods, for example, the cloud-NARX model [40], fuzzy model [49].

CRediT authorship contribution statement

Yuanlin Gu: Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Baihua Li:** Conceptualization, Resources, Writing - review & editing, Supervision, Project administration. **Qinggang Meng:** Conceptualization, Resources, Writing - review & editing, Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors are grateful to the Newton Fund for financial support under grant reference 104314.

References

[1] World Health Organization, (n.d.). http://www9.who.int/airpollution/en/.

[2] OECD, (n.d.). https://www.oecd.org/environment/indicators-modelling-outlooks/.

[3] J. Davidson, Beijing's Air Quality Continues to Show Significant Improvement, Eco Watch. (2020). https://www.ecowatch.com/beijing-air-pollution-2644261464.html.

[4] K. He, F. Yang, Y. Ma, Q. Zhang, X. Yao, C.K. Chan, S. Cadle, T. Chan, P. Mulawa, The characteristics of PM2.5 in Beijing, China, Atmos. Environ. 35 (29) (2001) 4959–4970, https://doi.org/10.1016/S1352-2310(01)00301-6.

[5] P.-H. Yang, M.-T. Hsieh, G.-M. Lin, M.-J. Chen, C.-H. Yeh, Z.-X. Huang, C.-M. Yang, Prediction of Outpatient Visits for Upper Respiratory Tract Infections by Machine Learning of PM2.5 and PM10 Levels in Taiwan, in: 2018 IEEE Int. Conf. Consum. Electron., 2018.

[6] P. Perez, J. Reyes, Prediction of particulate air pollution using neural techniques, Neural Comput. Appl. 10 (2001) 165–171, https://doi.org/10.1007/s005210170008.

[7] P. Pérez, A. Trier, J. Reyes, Prediction of PM2.5 concentrations several hours in advance using neural networks in Santiago, Chile, Atmos. Environ. 34 (8) (2000) 1189–1196, https://doi.org/10.1016/S1352-2310(99)00316-7.

[8] J.B. Ordieres, E.P. Vergara, R.S. Capuz, R.E. Salazar, Neural network prediction model for fine particulate matter (PM 2.5) on the US-Mexico border in El Paso (Texas) and Ciudad Juárez (Chihuahua), Environ. Model. Softw. 20 (5) (2005) 547–559, https://doi.org/10.1016/j.envsoft.2004.03.010.

[9] F. Biancofiore, M. Busilacchio, M. Verdecchia, B. Tomassetti, E. Aruffo, S. Bianco, S. Di Tommaso, C. Colangeli, G. Rosatelli, P. Di Carlo, Recursive neural network model for analysis and forecast of PM10 and PM2.5, Atmos. Pollut. Res. 8 (2017) 547–559, https://doi.org/10.1016/j.apr.2016.12.014.

[10] X.Y. Ni, H. Huang, W.P. Du, Relevance analysis and short-term prediction of PM2.5 concentrations in Beijing based on multi-source data, Atmos. Environ. 150 (2017) 146–161, https://doi.org/10.1016/j.atmosenv.2016.11.054.

[11] W.Z. Lu, H.Y. Fan, S.M. Lo, Application of evolutionary neural network method in predicting pollutant levels in downtown area of Hong Kong, Neurocomputing. 51 (2003) 387–400, https://doi.org/10.1016/S0925-2312(02)00623-9.

[12] C.J. Huang, P.H. Kuo, A deep CNN-LSTM model for particulate matter (PM2.5) forecasting in smart cities, Sensors (Switzerland) 18 (2018) 2220, https://doi.org/10.3390/s18072220.

[13] M. Krishan, S. Jha, J. Das, A. Singh, M.K. Goyal, C. Sekar, Air quality modelling using long short-term memory (LSTM) over NCT-Delhi, India, Air Qual. Atmos. Heal. 12 (8) (2019) 899–908, https://doi.org/10.1007/s11869-019-00696-7.

[14] W. Tong, L. Li, X. Zhou, A. Hamilton, K. Zhang, Deep learning PM2.5 concentrations with bidirectional LSTM RNN, Air Qual, Atmos. Heal. 12 (4) (2019) 411–423, https://doi.org/10.1007/s11869-018-0647-4.

[15] J. Wang, X. Zhang, Z. Guo, H. Lu, Developing an early-warning system for air quality prediction and assessment of cities in China, Expert Syst. Appl. 84 (2017) 102–116, https://doi.org/10.1016/j.eswa.2017.04.059.

[16] Y. Bai, B. Zeng, C. Li, J. Zhang, An ensemble long short-term memory neural network for hourly PM2.5 concentration forecasting, Chemosphere. 222 (2019) 286–294, https://doi.org/10.1016/j.chemosphere.2019.01.121.

[17] J. Fan, Q. Li, J. Hou, X. Feng, H. Karimian, S. Lin, A spatiotemporal prediction framework for air pollution based on deep RNN, ISPRS Ann, Photogramm. Remote Sens. Spat. Inf. Sci. 4 (2017) 15–22, https://doi.org/10.5194/isprs-annals-IV-4-W2-15-2017.

[18] W. Sun, H. Zhang, A. Palazoglu, A. Singh, W. Zhang, S. Liu, Prediction of 24-hour-average PM2.5 concentrations using a hidden Markov model with different emission distributions in Northern California, Sci. Total Environ. 443 (2013) 93–103. doi:10.1016/j.scitotenv.2012.10.070.

[19] Ming Dong, Dong Yang, Yan Kuang, David He, Serap Erdal, Donna Kenski, PM2.5 concentration prediction using hidden semi-Markov model-based times series data mining, Expert Syst. Appl. 36 (5) (2009) 9046–9055, https://doi.org/10.1016/j.eswa.2008.12.017.

[20] J. Wang, G. Song, A Deep Spatial-Temporal Ensemble Model for Air Quality Prediction, Neurocomputing. 314 (2018) 198–206, https://doi.org/10.1016/j.neucom.2018.06.049.

[21] W. Sun, J. Sun, Daily PM2.5 concentration prediction based on principal component analysis and LSSVM optimized by cuckoo search algorithm, J. Environ. Manage. 188 (2017) 144–152, https://doi.org/10.1016/j.jenvman.2016.12.011.

[22] B. Zhai, J. Chen, Development of a stacked ensemble model for forecasting and analyzing daily average PM2.5 concentrations in Beijing, China, Sci. Total Environ. 635 (2018) 644–658, https://doi.org/10.1016/j.scitotenv.2018.04.040.

[23] I. Bougoudis, K. Demertzis, L. Iliadis, V.-D. Anezakis, A. Papaleonidas, FuSSFFra, a fuzzy semi-supervised forecasting framework: the case of the air pollution in Athens, Neural Comput. Appl. 29 (7) (2018) 375–388, https://doi.org/10.1007/s00521-017-3125-2.

[24] B.M.G. Kibria, L. Sun, J.V. Zidek, N.D. Le, Bayesian spatial prediction of random space-time fields with application to mapping PM2.5 Exposure, J. Am. Stat. Assoc. 97 (457) (2002) 112–124, https://doi.org/10.1198/016214502753479275.

[25] S.K. Sahu, K.V. Mardia, A Bayesian kriged Kalman model for short-term forecasting of air pollution levels, J. R. Stat. Soc. Ser. C, Appl. Stat. 54 (1) (2005) 223–244, https://doi.org/10.1111/j.1467-9876.2005.00480.x.

[26] Q. Di, P. Koutrakis, J. Schwartz, A hybrid prediction model for PM2.5 mass and components using a chemical transport model and land use regression, Atmos. Environ. 131 (2016) 390–399, https://doi.org/10.1016/j.atmosenv.2016.02.002.

[27] P.D. Sampson, M. Richards, A.A. Szpiro, S. Bergen, L. Sheppard, T.V. Larson, J.D. Kaufman, A regionalized national universal kriging model using Partial Least Squares regression for estimating annual PM2.5 concentrations in epidemiology, Atmos. Environ. 75 (2013) 383–392, https://doi.org/10.1016/j.atmosenv.2013.04.015.

[28] M. Lee, M. Brauer, P. Wong, R. Tang, T.H. Tsui, C. Choi, W. Cheng, P.C. Lai, L. Tian, T.Q. Thach, R. Allen, B. Barratt, Land use regression modelling of air pollution in high density high rise cities: A case study in Hong Kong, Sci. Total Environ. 592 (2017) 306–315, https://doi.org/10.1016/j.scitotenv.2017.03.094.

[29] S.A. Billings, Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains (2013), https://doi.org/10.1002/9781118535561.

[30] S. Jeya, L. Sankari, Air Pollution Prediction by Deep Learning Model, in: Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020, 2020. doi:10.1109/ICICCS48265.2020.9120932.

[31] Qing Tao, Fang Liu, Yong Li, Denis Sidorov, Air Pollution Forecasting Using a Deep Learning Model Based on 1D Convnets and Bidirectional GRU, IEEE Access. 7 (2019) 76690–76698, https://doi.org/10.1109/Access.628763910.1109/ACCESS.2019.2921578.

[32] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion. 58 (2020) 82–115, https://doi.org/10.1016/j.inffus.2019.12.012.

[33] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should i trust you?" Explaining the predictions of any classifier, in: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2016. doi:10.1145/2939672.2939778.

[34] S.M. Lundberg, S.I. Lee, A unified approach to interpreting model predictions, in: Adv. Neural Inf. Process. Syst., 2017.

[35] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.R. Müller, Explaining nonlinear classification decisions with deep Taylor decomposition, Pattern Recognit. 65 (2017) 211–222, https://doi.org/10.1016/j.patcog.2016.11.008.

[36] M.D. Zeiler, G.W. Taylor, R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning, Proc. IEEE Int. Conf. Comput. Vis. (2011), https://doi.org/10.1109/ICCV.2011.6126474.

[37] V. Krakovna, F. Doshi-Velez, Increasing the Interpretability of Recurrent Neural Networks Using Hidden Markov Models, ArXiv Prepr. (2016).

[38] S. Chen, S.A. Billings, Representations of non-linear systems: the NARMAX model, Int. J. Control. 49 (3) (1989) 1013–1032, https://doi.org/10.1080/00207178908559683.

[39] H.L. Wei, S.A. Billings, Model Structure Selection Using an Integrated Foward Orthogonal Search Algorithm Assisted by Square Correlation and Mutual Information, Int. J. Model. Identif. Control. 3 (2008) 341–356, https://doi.org/10.1504/IJMIC.2008.020543.

[40] Y. Gu, Hua-Liang Wei, Richard J. Boynton, Simon N. Walker, Michael A. Balikhin, System Identification and Data-Driven Forecasting of AE Index and Prediction Uncertainty Analysis Using a New Cloud-NARX Model, J. Geophys. Res. Sp. Phys. 124 (1) (2019) 248–263, https://doi.org/10.1029/2018JA025957.

[41] Y. Gu, H.L. Wei, A robust model structure selection method for small sample size and multiple datasets problems, Inf. Sci. (Ny) 451–452 (2018) 195–209, https://doi.org/10.1016/j.ins.2018.04.007.

[42] G.R. Bigg, H.L. Wei, D.J. Wilton, Y. Zhao, S.A. Billings, E. Hanna, V. Kadirkamanathan, A century of variation in the dependence of Greenland iceberg calving on ice sheet surface mass balance and regional climate change, Proc. R. Soc. A Math. Phys. Eng. Sci. 470 (2166) (2014) 20130662, https://doi.org/10.1098/rspa.2013.0662.

[43] Yuanlin Gu, Yuan Yang, Julius P.A. Dewald, Frans C.T. van der Helm, Alfred C. Schouten, Hua-Liang Wei, Nonlinear Modeling of Cortical Responses to Mechanical Wrist Perturbations using the NARMAX Method, IEEE Trans. Biomed. Eng. 68 (3) (2021) 948–958, https://doi.org/10.1109/TBME.1010.1109/TBME.2020.3013545.

[44] S.A. Billings, H.L. Wei, An adaptive orthogonal search algorithm for model subset selection and non-linear system identification, Int. J. Control. 81 (5) (2008) 714–724, https://doi.org/10.1080/00207170701216311.

[45] Xinghan Xu, Minoru Yoneda, Multitask Air-Quality Prediction Based on LSTM-Autoencoder Model, IEEE Trans. Cybern. 51 (5) (2021) 2577–2586, https://doi.org/10.1109/TCYB.2019.2945999.

[46] UC Irvine Machine Learning Repository, (n.d.). https://archive.ics.uci.edu/ml/index.

[47] Y. Gu, H.-L. Wei, Significant Indicators and Determinants of Happiness: Evidence from a UK Survey and Revealed by a Data-Driven Systems Modelling Approach, Soc. Sci. 7 (2018) 53, https://doi.org/10.3390/socsci7040053.

[48] T.E. Akinola, E. Oko, Y. Gu, H. Wei, M. Wang, Non-linear system identification of solvent-based post-combustion CO2 capture process, Fuel. 239 (2019) 1213–1223.

[49] C. He, M. Mahfouf, L.A. Torres-Salomao, An Adaptive General Type-2 Fuzzy Logic Approach for Psychophysiological State Modeling in Real-Time Human-Machine Interfaces, IEEE Trans. Human-Machine Syst. 51 (1) (2021) 1–11, https://doi.org/10.1109/THMS.2020.3027531.

Yuanlin Gu received the B.S. degree in automatic control from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013, and the M.S. degree in control systems from University of Sheffield, Sheffield, UK, in 2014, and the PhD degree in data modelling and data mining from University of Sheffield, Sheffield, UK, in 2019. He is a Lecturer in Computer Science with University of Roehampton, London, UK. He has worked at Loughborough University before he joined the University of Roehampton. His research interests include data modelling and identification for complex nonlinear systems, interpretable machine learning and deep learning, and uncertainty analysis for time series prediction.

Baihua Li received the B.S. and M.S. degrees in electronic engineering from Tianjin University, Tianjin, China, in 1989 and 1994, and the Ph.D. degree in computer science from Aberystwyth University, Aberystwyth, U.K., in 2003. She is a Professor with the Department of Computer Science, Loughborough University, U.K. She has worked at Tianjin University and Manchester Metropolitan University before she joined the Department of Computer Science, Loughborough University. Her research interests include innovations and novel applications of internet of things, computer vision, and pattern recognition techniques in various fields. She has published more than 50 papers in high impact journals and conferences of international standard.

Qinggang Meng received the B.S. and M.S. degrees from the School of Electronic Information Engineering, Tianjin University, China, and the Ph.D. degree in computer science from Aberystwyth University, U.K. He is a Professor with the Department of Computer Science, Loughborough University, U.K. His research interests include biologically and psychologically inspired learning algorithms and developmental robotics, service robotics, robot learning and adaptation, multi-UAV cooperation, drivers distraction detection, human motion analysis and activity recognition, activity pattern detection, pattern recognition, artificial intelligence, and computer vision. He is a fellow of the Higher Education Academy, U.K.