Temporal Spatial Decomposition and Fusion Network for Time Series Forecasting*

1st Liwang Zhou
Zhejiang University
China
21731005@zju.edu.cn

2nd Jing Gao
Anhui University
China
jingles980@gmail.com

Abstract—Feature engineering is required to obtain better results for time series forecasting, and decomposition is a crucial one. One decomposition approach often cannot be used for numerous forecasting tasks since the standard time series decomposition lacks flexibility and robustness. Traditional feature selection relies heavily on preexisting domain knowledge, has no generic methodology, and requires a lot of labor. However, most time series prediction models based on deep learning typically suffer from interpretability issues, so the "black box" results lead to a lack of confidence. Dealing with the above matters forms the motivation of the thesis. In the paper, we propose TSDFNet as a neural network with a self-decomposition mechanism and an attentive feature fusion mechanism. It abandons feature engineering as a preprocessing convention and creatively integrates it as an internal module with the deep model. The self-decomposition mechanism empowers TSDFNet with extensible and adaptive decomposition capabilities for any time series and is more effective in predicting problems than traditional decomposition methods. Users can choose their own basis functions to decompose the sequence into temporal and generalized spatial dimensions. The attentive feature fusion mechanism has the ability to capture the importance of external variables and the causality with target variables. It can automatically suppress the unimportant features while enhancing the effective ones so that users do not have to struggle with feature selection. The basic idea of TDFNNet is straightforward. It integrates feature decomposition, feature selection, and prediction into one model. Moreover, TSDFNet can quickly look into the "black box" of the deep neural network by feature visualization and analyzing the prediction results. We demonstrate performance improvements over existing widely accepted models on more than a dozen datasets, and three experiments showcase the interpretability of TSDFNet.

Index Terms—time series, interpretability, long-term prediction, deep learning

I. INTRODUCTION

Time series forecasting plays a crucial role in numerous fields such as economy [1], finance [2], transportation [3], meteorology [4]. It empowers people to foresee opportunities and serves as guidance for decision-making. Therefore, it is crucial to increase the generality of time series models and lower modeling complexity while maintaining performance. In the field of time series forecasting, multi-variable and multi-step forecasting forms one of the most challenging tasks. Errors may accumulate as the forecast step increases. At present, there is no universal method to handle the problem of multi-variable and multi-step time series prediction. One time series usually calls for its specific feature engineering

*This work was supported by the National Natural Science Foundation of China (Grant No. 61803163) and the Natural Science Foundation of Zhejiang Province, China (Grant No. LY19F030005).

• We proposed an Attentive Feature Fusion Network and forecasting model due to the complexity and diversity of real-world time series, which generally requires data analysts to have specialized background knowledge.

Feature engineering is usually used to preprocess data before modeling. In the field of feature engineering, time series decomposition is a classical method to decompose a complex time series into numerous predictable sub-series, such as STL [31] with seasonal and trend decomposition, EEMD [19] with ensemble empirical mode decomposition, EWT [20] with empirical wavelet transform. Although these methods have the adaptive ability, they have determined the decomposition way before model training, and the pre-decomposed subsequence may not be helpful for prediction. In addition, feature selection is another crucial step. For complex tasks, some auxiliary variables are usually needed to assist in the prediction of target variables. The reasonable selection of additional features is essential to the model’s performance because introducing some additional redundant features may degrade the model’s performance. Choosing the appropriate decomposition methods and important additional features is also a challenging problem for data analysts.

Therefore, it is necessary to break the traditional practice and devise a new way to handle these problems. This study developed a novel neural network model called TSDFNet based on the self-decomposition and attentive feature fusing mechanisms. Decomposition and feature selection are integrated as internal modules of the deep model to lessen complexity and increase adaptability. Moreover, it is more convenient and performs better than the traditional preprocessing method. In summary, the contributions are summarized as follows:

• We designed a self-decomposition network that includes a Temporal Decomposition Network (TDN) and Spatial Decomposition Network (SDN). It can optimize the decomposition coefficient according to the prediction error to choose the most reasonable decomposition method, which is more effective in sequence prediction than the traditional decomposition method. TDN decomposes time series over temporal dimensions and allows users to customize basis functions for specific tasks. SDN creatively uses high-dimensional external features as decomposition basis functions to model the relationship between external and target variables.

• We proposed an Attentive Feature Fusion Network...
AFFN), which has the ability to automatically feature selection and capture features’ importance and causality. In this way, users can avoid the trouble of feature selection and use arbitrary basis functions in the self-decomposition network without worrying about the loss of model performance caused by introducing invalid features.

- Our proposed approach creatively integrates sequence decomposition, feature selection, and prediction into an end2end model. TSDFNet obtains interpretable results on datasets in multiple fields and has significantly improved performance compared with many traditional models.

II. RELATED WORK

The field of time series prediction has a rich history, and many outstanding models have been developed. The most well-known conventional methods include ARIMA [6] and exponential smoothing [7]. The interpretability and usability of the ARIMA model, which turns nonstationary processes into stationary ones through difference, can also be further expanded into VAR [8] to address the issue of multivariate time series forecasting, which are the main reasons for its popularity.

In recent years, deep learning has become famous and neural networks have achieved success in many fields [23], [24], [25]. It uses the backpropagation algorithm [26] to optimize the network parameters. Long Short-Term Memory (LSTM) [10], and its derivatives show great power in sequential data. It overcomes the defect of vanishing gradient of recurrent neural network (RNN) [11] and can better capture long-term dependence. Sequence to Sequence (Seq2Seq) [12] usually uses a pair of LSTMs, or GRUs [13] as encoder and decoder. The encoder maps the input data to the hidden space into a fixed-length semantic vector. The decoder reads the context vector and attempts to predict the variable target step by step. Temporal convolutional network (TCN) [14] could also be effectively applied to the sequence prediction problem, which may be used as an alternative to the popular RNN family of methods and has faster speed and fewer parameters compared with RNN-based models with causal convolution and residual connection. Some models combine the advantages of CNN and RNN, using CNN to extract short-term local relationship dependence patterns between variables and using RNN to discover long-term patterns of time series trends. A typical one is Lstnet [29], which uses the traditional autoregressive model to solve the problem of scale insensitivity of neural network models. The attention mechanism [17] emerged as an improvement over the encoder-decoder based [18], and it can easily be further extended into a self-attentional mechanism as the core of the Transformer models [15], [16].

III. METHODOLOGY

The network’s architecture is depicted in Figure 1. It has two main parts, the first of which is a self-decomposition network that includes TDN and SDN. The feature fusion network (AFFN) is another part. Self-decomposition networks decompose sequences into more predictable subsequences that participate in the prediction along with exogenous variables as inputs to AFFN.

- Fig. 1: Overall structure of TSDFNet

A. Self-decomposing network

The structure of the self-decomposition network includes two decomposition modules. One is the time decomposition network TDN, which adopts custom basis functions to decompose sequences in the time dimension. The other is the spatial decomposition network SDN, which decomposes sequences in generalized spatial dimensions using exogenous features as essential functions. Its main objective is to break down complex sequences into simple and predictable ones.

TDN uses multiple sets of pre-trained basis functions with different parameters to capture signal features, which could be Fourier basis functions, polynomial basis functions, wavelet basis functions, and so on.

- Fig. 2: Temporal Decomposition Network

The architecture of TDN is shown in Figure 2. There are $N$ recursive decomposition units in it. $(n + 1)^{th}$ unit accepts its respective input $X_n$ as input and output two intermediate
components $W_n$ and $V_n$. Each decomposition unit consists of two parts. Stacked fully connected network $L_s$ maps data into a hidden space to produce the semantic vector $S_n$, predicts basis expansion coefficients both forward and backward through two sets of fully connected networks $L_P$ and $L_Q$ respectively. The process is:

$$S_n = L_s(X_n)$$  \hspace{1cm} (1)

$$P_n = L_P(S_n)$$ \hspace{1cm} (2)

$$Q_n = L_Q(S_n)$$ \hspace{1cm} (3)

The other part is a group of pretrained basis function models, which are functions of the time vector $t = [-w, -w + 1, \ldots, 0, h - 1, h/L]$ defined on a linear space from $-w/L$ to $h/L$, where $L = w + h + 1, w$ is the time historical window length of drive sequence, $h$ is the time window length of the target sequence. This time vector is fed into different pre-trained models and mapped into multiple basis functions, such as trigonometric functions with different frequencies defined by $C_n = [\sin(-kt), \cos(-kt), \ldots, \cos(kt), \sin(kt)]$, polynomial function with different degrees defined by $C_n = [t, t^2, t^3 \ldots t^k]$. $C_n$ is divided into $C_n^p$ when $t = [-w/L, \ldots, 0]$ and $C_n^q$ when $t = [0, h/L]$, which are used to fit historical and future data respectively. Their coefficient matrix $P_n$ and $Q_n$ determine the importance of each basis function. The final outputs of $n^{th}$ block are defined by:

$$W_n = C_n^pP_n$$ \hspace{1cm} (4)

$$V_n = C_n^qQ_n$$ \hspace{1cm} (5)

The input to the next block is defined as follows:

$$X_{n+1} = X_n - W_{n+1}$$ \hspace{1cm} (6)

The original signal $X_0 = [x_{-w}, x_{-w+1}, \ldots, x_{t-1}]$ keeps removing backward feature components $W_n$, and ideally the residual is a random noise that no longer contains feature information. The process is:

$$\text{residual} = X_0 - \sum_{i=0}^{n+1} W_i$$ \hspace{1cm} (7)

The forward output of each decomposition unit is accumulated to form the TDN output as follow:

$$V = \sum_{i=0}^{n+1} V_i$$ \hspace{1cm} (8)

The model’s hyperparameter $N$, which is defined as the number of decomposition layers, is dependent on the kind of basis function you select and is repeatable for each basis function. The weights of these basis function networks can be adjusted once again for different tasks to adapt to diverse scenarios. This means that $C_n$ is not fixed, it will be independent learning into other comparable forms.

The spatial decomposition network (SDN) is shown in Figure 3. Its structure is similar to TDN. The difference is that SDN adopts external features mapped to higher dimensions as the basis vectors $C_n^p$ and $C_n^q$. Details are shown as follows:

$$C_n^p = L_n(E_p)$$ \hspace{1cm} (9)

$$C_n^q = L_n(E_q)$$ \hspace{1cm} (10)

where $E_p$ is the historical additional feature and $E_q$ is the future additional feature. $L_n$ are stacked fully connected modules that map historical and future additional features to higher dimensions, respectively. We employ embedding to transform some discrete features and use 0 to fill in the missing features. Additionally, it is important to note that the self-decomposition module offers flexibility to handle the univariable time series. The module of the spatial decomposition network (SDN) can be disabled, or alternatively, SDN can be enabled and accept feature components of other methods as input, such as EMD [28].
B. Attentive Feature Fusion network

Attentive feature fusion network, as shown in Figure 4, the feature selection module is designed to provide instance-wise variable selection, groups of decision units iteratively suppress irrelevant features, the multi-head attention block accepts the result of the feature selection modules as input to further model global relationships. Finally, the output is obtained by the GLU (Gated Linear Unit) [33] followed by FC (Fully Connected) at each time step. Here GLU attempts to suppress the invalid part of the input data.

A traditional method to judge whether a feature is essential to the prediction is usually to add or subtract the input of the feature to observe the performance change. This approach can be cumbersome if the number of features is too large. And it cannot rule out cross-effects between features. So the basic idea of AFFN is to simulate the above process control using a neural network and recursively control the input of features by a series of learnable masks. The final output is a superposition of the results of different feature selection modules. We can visualize the importance of features by analyzing these learnable masks.

\[
D = \frac{1}{B \times T \times J} \sum_{b=1}^{B} \sum_{t=0}^{T} \sum_{j=1}^{J} M_{b,t,j} \tag{11}
\]

The feature selector has shared layers composed of a GLU, LN [22]. Layer normalization (LN) is a technique to normalize the distributions of intermediate layers. It enables smoother gradients, faster training, and better generalization accuracy. The output of share layers \( s \) in feature selector is divide into two parts \( [s_1, s_2] \), one of them \( f_1(s_1) \) are used as input of mask generator, another \( f_2(s_2) \) is as output of \( j^{th} \) level of decision, where \( f_1 \) is Fully Connected layer, \( f_2 \) is Relu, if the element in \( s_2 < 0 \), it has no contribution to current decision output.

Multiple attention was initially applied to language models, and we found that integrating it helped both the interpretation and performance of the model. Its details are shown as follows:

\[
A(Q, K) = \text{Softmax}(QK^T / \sqrt{d_{attn}}) \tag{12}
\]

\[
\text{Attention}(Q, K, V) = A(Q, K)V \tag{13}
\]

\[
\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{14}
\]

\[
\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \tag{15}
\]

Among them, \( \text{MultiHead}(Q, K, V) \) represents the Attention function. \( \text{Softmax} \) is the probability distribution function, and a parameter \( d_{attn} \) used to normalize features on a scale. In this model, \( Q \) is the high-dimensional feature of all historical information output from the feature selection module, while \( K \) and \( V \) are the high-dimensional feature of future known information.

IV. EXPERIMENT

To evaluate the proposed TSDFNet, we chose several typical models, including TCN based on causal convolution, Lstnet [29] based on CNN and RNN, Seq2Seq based on attention and LSTM, Lstm-SAE [30] based on stacked autoencoder, for thorough comparison on various types of datasets. These models are chosen because they are representative and widely used in various data science competitions.

A. Implementation details

The experiment is based on the Pytorch framework in the Ubuntu system. Hardware configuration is Intel(R) Core(TM) i7-6800K CPU @ 3.40 GHZ, 64 GB memory, GeForce GTX 1080 GPU.

First, we construct several sets of synthetic basis functions with different parameters \( k \), which provides diversity to model \( y_i = C_i(kt) \), such as trigonometric functions, polynomial functions, and so on. These models are a three-layer, fully connected network trained with L2 loss. Each dataset has 10000 training samples, the training period is 100, and the batch size in the experiment is set to 40.

The second step is to train the TSDFNet using the ADAM [27] optimizer with an initial learning rate of 0.0001. The
dropout rate is set to 0.1 for better generalization, and the batch size is set to 32. Early stopping is employed to avoid overfitting. The training process will be terminated if there is no loss degradation within ten epochs. Metrics were RAE and SMAPE defined by

$$RAE = \frac{\sum_{(i,t) \in \Omega_{test}} |Y_{it} - \hat{Y}_{it}|}{\sum_{(i,t) \in \Omega_{test}} |Y_{it} - \text{mean}(Y)|}$$ (16)

$$SMAPE = \frac{2 \sum_{(i,t) \in \Omega_{test}} |Y_{it} - \hat{Y}_{it}|}{N \sum_{(i,t) \in \Omega_{test}} |Y_{it}| + |\hat{Y}_{it}|}$$ (17)

B. Univariate time series prediction and result analysis

We list the univariate results of five typical datasets in Table I and Table VII.

**ECG5000** A total of 5000 electrocardiograms (ECGs), 4500 samples for training and 500 samples for testing, make up the dataset, which is derived from the UCR time series. Each sample in the sequence is 140-time points. The experiment uses the first 84 time steps as input, while the final 56 time steps are predicted as a result. The result is shown in Figure 6. Although the information came from different patients and included abnormal data, it was generally regular. The data in the first half of the cycle are relatively stable, and there are peaks and troughs in the second half of the process, which is challenging to forecast. Compared with other methods, the prediction accuracy of TSDFNet is the highest, and the waveform predicted by TSDFNet best fits the real situation.

All models accurately predict a decline in the evening, but overall, TSDFNet’s SMAPE and RAE are minor.

**ItalyPowerDemand** The dataset is derived from a 12-month time series of Italy’s power demand. 67 samples were evaluated after 1029 samples had been trained. There are 24 samples in total in each set. In the experiment, the data from the first 18 hours are utilized as input to forecast the data from the following 6 hours. The result is shown in Figure 7. This household’s daily energy consumption data is also regular, with peaks in both morning and evening energy consumption.

**Retail** The dataset from Kaggle provides US monthly retail sales data from January 1, 1992, to May 1, 2016. It has only a total of 293 samples. 95 percent of the data were used as the training set and 5 percent as the test set in this paper. In order to forecast the data for the following six months, the data from the preceding 12 months were used as input. The result is shown in Figure 8. Retail sales rose month on month and fluctuated regularly. We can observe from the figure that TSDFNet’s prediction of details within a period is more accurate than that of other models, especially the parts with sharp changes. This experiment shows that TSDFNet has advantages over other methods in predicting trends combined with complex periodic type data, which is useful in practice.

**Airpassenger** The dataset provides monthly passenger counts for American Airlines on foreign airlines from 1949 to 1960. There are 144 samples in all. The training set consisted of data from 1949 to 1959, while the test set comprised data from 1960. The input data spans 60 months, and a 12-month projection is made for the data. The result is shown in Figure
We can observe from the figure that Seq2Seq almost failed in its prediction, while TSDFNet had the highest accuracy and made well-detailed predictions. The other methods only roughly capture the characteristics of the data. This experiment demonstrates that TSDFNet can also perform well on small sample data sets.

The Dataset was obtained from Kaggle, which spans the years 2006 to 2016. The experiment’s temperature prediction is dependent on other variables such as humidity, wind speed, wind direction, visibility, cloud cover, air pressure, etc. A total of 96453 samples were included in the data, which were split up into hours. 95 percent of the data were utilized for training, and 5 percent were used for testing. The model predicted the temperature for the following 360 hours using data from the first 2,200 hours and the next 360 hours on external features. The result is shown in Figure 11. It can be observed that other models can only predict the approximate temperature value, while TSDFNet accurately indicates the temperature drop and rise.

### TABLE I: RAE of univariate time series models

|          | Seq2Seq | LstNet | TCN   | SAE   | Ours  |
|----------|---------|--------|-------|-------|-------|
| ECG5000  | 0.360   | 0.669  | 0.793 | 0.938 | 0.290 |
| Power    | 0.294   | 0.276  | 0.718 | 0.425 | 0.162 |
| Retail   | 1.803   | 1.536  | 0.855 | 1.426 | 0.201 |
| AirPassenger | 1.233 | 0.993  | 1.911 | 1.342 | 0.252 |
| Sunspot  | 0.644   | 0.904  | 0.671 | 0.743 | 0.334 |

### TABLE II: SMAPE of univariate time series models

|          | Seq2Seq | LstNet | TCN   | SAE   | Ours  |
|----------|---------|--------|-------|-------|-------|
| ECG5000  | 0.424   | 0.685  | 0.682 | 0.731 | 0.373 |
| Power    | 0.334   | 0.364  | 0.606 | 0.375 | 0.144 |
| Retail   | 0.052   | 0.115  | 0.098 | 0.158 | 0.013 |
| AirPassenger | 0.132 | 0.086  | 0.213 | 0.282 | 0.027 |
| Sunspot  | 0.577   | 0.847  | 0.573 | 0.757 | 0.431 |

### C. Multivariate time series prediction and result analysis

We list the multivariable results of five typical datasets in Table III and Table IV.

**Weather** The Dataset was obtained from Kaggle, which spans the years 2006 to 2016. The experiment’s temperature prediction is dependent on other variables such as humidity, wind speed, wind direction, visibility, cloud cover, air pressure, etc. A total of 96453 samples were included in the data, which were split up into hours. 95 percent of the data were utilized for training, and 5 percent were used for testing. The model predicted the temperature for the following 360 hours using data from the first 2,200 hours and the next 360 hours on external features. The result is shown in Figure 11. It can be observed that other models can only predict the approximate temperature value, while TSDFNet accurately indicates the temperature drop and rise.

### Lorenz

A 90-dimensional coupled Lorenz model $X'(t) = \ldots$
\( G(X(t); \mathbf{P}) \) is used to generate the synthesis time data set under different noise conditions, where \( G(\cdot) \) is the nonlinear function set of the Lorenz system with \( X(t) = (x_1, \ldots, x_9) \), \( \mathbf{P} \) is the parameter vector. To demonstrate the distinction between the model described in this work in the time-varying system and the time-invariant system, the time-invariant Lorenz system and the time-varying Lorenz system are tested in this article, respectively. For a time-invariant Lorenz system (Lorenz-S), \( \mathbf{P} \) does not change over time; however, for time-varying systems (Lorenz-D), \( \mathbf{P} \) does. In the experiment simulation, 5000 samples were generated; the first 90% of the data was utilized as the training set and the final 10% as the test set. The experimental findings demonstrate that TSDFNet works well effectively in both time-varying and time-invariant systems. The result is shown in Figure 12 and Figure 13.

**Figure 12:** Forecast results of time invariant lorenz system

**Figure 13:** Forecast results of time variant lorenz system

**Windspeed** This dataset, which is a high-dimensional (155-dimensional) windspeed data set from 155 stations in Wakkanai, Japan, is supplied by the Japan Meteorological Agency. Every 10 minutes, for a total of 138,600 minutes, are recorded. The known duration of 110 minutes is used in this paper to estimate the wind speed for the subsequent 40 minutes. Data from 154 other sites were used as additional features, and a target wind speed at one of 155 sites was chosen at random. 95% of the initial data was utilized as a training set, while 5% was used as a test set. Even though wind speed is typically thought to be exceedingly challenging to predict, this approach predicts outcomes that are superior to those anticipated by other methods. The result is shown in Figure 14. The performance of TSDFNet is significantly better than that of the other methods. This result shows the great advantage of TSDFNet in its high robustness because it works well almost everywhere within 138,600 min, considering that multistep-ahead wind speed prediction is usually a difficult task.

**Traffic** The average vehicle speed (MPH) is predicted using data from 207 loop detectors along Route 134 in Los Angeles, with the observations from each detector being handled as a separate variable. The first 120 hours are utilized to forecast the 24-hour average speed. The high-dimensional data from one sensor is chosen as the target variable, while the observations from other sensors are used as additional features. The result is shown in Figure 15. Clearly, this result indicates that TSDFNet accurately predicted the dynamical behaviors for non-periodic and highly-fluctuating cases based on only short-term data and demonstrates the model is capable of predicting spatial information.

**D. Interpretability Results**

After establishing the performance advantages of TSDFNet, we then demonstrate how our model design allows the analysis of its specific components to explain the general relationships it has learned.

We first quantify feature importance by analyzing the interpretable variants of TDN described in Eq. (4). The air passenger dataset is not stationary and combines seasonality
and trend. It grows yearly and frequently varies from month to month. The outcome is depicted in Figure 16. The original signal and its prediction are displayed in the figure's top row, while the basis function is shown in the second row.

The outcome for the sunspot dataset is displayed in Figure 17. The top row of the graphic makes evident how seasonal it is, while the bottom row demonstrates how successfully periodicity is captured by the fine-tuned trigonometric basis functions. Additionally, attention weight patterns, which highlight the crucial moments the TSDFNet based its inferences on, might be quite helpful. We can see many brilliant stripes in the second row, spaced out against a dark background. They stand for the start of a fresh cycle. It can be seen that the attention spikes match the troughs of the sunspot when the selection weights for each time step are aggregated, as illustrated in the third row.

Additionally, the importance of each feature, as determined by (11), can be examined using the decision weight patterns in the feature selection block. The importance distribution of historical features is depicted in Figure 18 while also accounting for other feature components of historical data. We can see that previous temperature data account for around 75% of the importance of future temperature data. The importance distribution of upcoming known data is depicted in Figure 19. The accuracy of the predictions depends heavily on the air pressure and cloud thickness of the future. The contribution to the result is also zero because the future temperature data is unknowable and filled with a value of 0. We also use the Gaussian noise as input to show how our model may exclude unimportant variables. Gaussian noise’s contribution is minimal because it doesn’t contain relevant information.

### TABLE III: RAE of multivariate time series models

|       | Seq2Seq | Lstnet | TCN  | SAE  | Ours |
|-------|---------|--------|------|------|------|
| Lorenz-S | 0.202   | 0.109  | 0.474| 0.685| **0.060** |
| Lorenz-D | 0.888   | 1.174  | 1.574| 1.230| **0.624** |
| Weather  | 0.305   | 1.985  | 1.822| 1.204| **0.218** |
| Windspeed| 0.294   | 0.428  | 0.531| 0.284| **0.094** |
| Traffic  | 0.262   | 0.241  | 0.241| 0.588| **0.074** |

### TABLE IV: SMAPE of multivariable time series models

|       | Seq2Seq | Lstnet | TCN  | SAE  | Ours |
|-------|---------|--------|------|------|------|
| Lorenz-S | 0.448   | 0.216  | 0.517| 0.487| **0.101** |
| Lorenz-D | 0.785   | 1.292  | 2.675| 1.080| **0.675** |
| Weather  | 5.151   | 7.338  | 4.565| 4.426| **0.899** |
| Windspeed| 3.254   | 5.042  | 6.442| 1.468| **0.624** |
| Traffic  | 5.156   | 5.338  | 5.189| 5.741| **0.899** |

Fig. 15: Forecast results of traffic

Fig. 16: feature analysis of basis function of American air passenger

Fig. 17: Visualization of sunspot features
E. Ablation Analysis

To quantify the benefits of each of our proposed architectural contributions, ablation experiments are conducted. We ablate by separating the TDN, SDN, Feature Selection Network, and Multihead Attention layer with a simple linear layer and quantifying the percentage increase in Smape versus the original architecture.

In general, TDN dramatically impacts the performance of univariable data, while SDN and FSN contribute more benefits to the performance of multivariate data. The standard Multihead attention slightly improves the performance.

Experiments on weather datasets show that the adopted TDN can bring an average of 12.15% improvement. Although weather datasets are affected by many external factors, periodicity is still a noticeable feature, so TDN also plays a crucial role in predicting results.

We next attempt to verify the effectiveness of our self-decomposition architecture further. We removed the self-decomposition block and adopted more well-established decomposition algorithms as the pre-processing separately. Our proposed self-decomposition architecture consistently achieves superior performance.

TABLE V: Increase in SMAPE of univariable dataset by Ablation

|                  | TDN (%) | SDN (%) | FSN (%) | Attention (%) |
|------------------|---------|---------|---------|---------------|
| ECG5000          | 11.22%  | 2.16%   | 2.25%   | 1.21%         |
| Power            | 13.58%  | 1.32%   | 3.25%   | 2.15%         |
| Retail           | 22.15%  | 2.58%   | 1.32%   | 2.26%         |
| AirPassenger     | 20.14%  | 0.04%   | 1.24%   | 1.68%         |
| Sunspot          | 18.25%  | 1.38%   | 1.19%   | 5.74%         |

TABLE VI: Increase in SMAPE of multivariable dataset by Ablation

|                  | TDN (%) | SDN (%) | FSN (%) | Attention (%) |
|------------------|---------|---------|---------|---------------|
| Lorenz-S         | 6.22%   | 12.28%  | 11.13%  | 3.21%         |
| Lorenz-D         | 3.78%   | 11.49%  | 12.67%  | 1.08%         |
| Weather          | 12.15%  | 6.23%   | 9.56%   | 4.26%         |
| Windspeed        | 3.24%   | 10.04%  | 16.44%  | 1.68%         |
| Traffic          | 5.15%   | 15.38%  | 11.19%  | 2.74%         |

F. Basis Function Selection Policy

Selecting custom basis functions is the key to effectively representing the signal. Although using more components allows us to best preserve the historical information in the time series, it may potentially lead to overfitting of the historical data and, consequently, a poor prediction of future signals. Hence, we need to select a subset of basis functions. In the experiment, we use the Fourier basis function to model periodicity and the polynomial basis function to model trend. For most tasks, these two basis functions are powerful enough to represent the signal, as demonstrated in previous experiments. Excellent performance can be achieved even in chaotic data without obvious periodicity and trends. In most cases, random Fourier basis function selection is a better policy in forecasting tasks, which requires no prior knowledge of the input and generalizes easily in new tasks. However, the fixed selection of low-order polynomial basis function policy is more effective for trend modeling because the growth trend of most data is relatively slow. Figure 20 compares fixed and random selection policies. It can be observed that random Fourier basis function selection and fixed polynomial basis function selection achieve better performance than other policies.

TABLE VII: Ablation of self-decomposition architecture

|                  | STL     | EMD     | EEMD    | EWT     | Ours    |
|------------------|---------|---------|---------|---------|---------|
| ECG5000          | 0.658   | 0.545   | 0.522   | 0.531   | 0.373   |
| Power            | 0.324   | 0.254   | 0.206   | 0.235   | 0.144   |
| Retail           | 0.032   | 0.025   | 0.023   | 0.027   | 0.013   |
| AirPassenger     | 0.062   | 0.046   | 0.044   | 0.073   | 0.027   |
| Sunspot          | 0.724   | 0.622   | 0.524   | 0.531   | 0.431   |
state-of-the-art performance compared to various well-known experiments demonstrate that the TSDFNet yields consistent selection, and prediction into one model. Finally, extensive extremely simple: to integrate feature decomposition, feature selects the most important ones. The basic idea of our design is feature Fusion Network (AFFN) fuses all the input features and as basis functions for sequence decomposition. Attentive Feature Spatial Decomposition Network (SDN) uses external features functions as the eigenmodes of time series decomposition, ing problem of time series, where Temporal Decomposition learning model which incorporates feature engineering into Fusion Network (TSDFNet) – a novel interpretable deep

V. CONCLUSION
This paper introduces the Temporal Spatial Decomposition Fusion Network (TSDFNet) – a novel interpretable deep learning model which incorporates feature engineering into modeling and achieves incredible performance. TSDFNet utilizes specialized components to handle the long-term forecasting problem of time series, where Temporal Decomposition Network (TDN) allows the user to customize arbitrary basis functions as the eigenmodes of time series decomposition, Spatial Decomposition Network (SDN) uses external features as basis functions for sequence decomposition. Attentive Feature Fusion Network (AFFN) fuses all the input features and selects the most important ones. The basic idea of our design is extremely simple; to integrate feature decomposition, feature selection, and prediction into one model. Finally, extensive experiments demonstrate that the TSDFNet yields consistent state-of-the-art performance compared to various well-known algorithms.

REFERENCES
[1] D. Edewusi, G. Ajayi, I. Ezerekel, “The Nexus between Tax Revenue and Economic growth in Nigeria,” International Journal of Applied Economics, Finance and Accounting, vol. 4, no. 2, pp. 45-55, 2019.
[2] Mohammad Obaidur Rahman, Md. Sabir Hossain, Ta-Seen Junaid, Md. Shafiu Alam Forhad, Muhammad Kamal Hossen, 2019. “Predicting Prices of Stock Market using Gated Recurrent Units (GRUs) Neural Networks,” IJCSNS, vol. 19 no. 1 pp. 213-222, 2019.
[3] S. Ishak and H. Al-Deek, “Performance evaluation of short-term time-series traffic prediction model,” Journal of transportation engineering, vol. 128, no. 6, pp. 490-498, Nov. Dec. 2002.
[4] Z. Liu, P. Jiang, L. Zhang, and X. Niu, “A combined forecasting model for time series: Application to short-term wind speed forecasting,” Applied Energy, vol. 259, Art. no. 114137, Feb. 2020.
[5] G. E. P. Box and Jenkins G M, “Time series analysis : Forecasting and control,” Journal of Forecasting, vol. 3, no. 3, 2010.
[6] E. D. McKenzie, “ General exponential smoothing and the equivalent arma process,” Journal of Forecasting, vol. 3, no. 3, pp. 333-344, 1984.
[7] C. Holt, “Forecasting seasonal and trends by exponentially weighted moving averages,” Int. J. Forecast, vol. 20, no. 1, pp. 5–10, Jan. –Mar. 2004.
[8] I. Melynk and A. Banerjee, “Estimating structured vector autoregressive model,” arXiv:1602.06606, 2016.
[9] K. J. Kim, “Financial time series forecasting using support vector machines,” Neurocomputing, vol. 55, pp. 307–319, 2003.
[10] S. Hochreiter and J. Schmidhuber, Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[11] Connor, L. E. Atlas, D. R. Martin. “Recurrent networks and NARMA modeling,” Advances in neural information processing systems, vol. 4, pp. 301-308, 1992.
[12] Sutskever, O. Vinyals, et. al, “Sequence to sequence learning with neural networks,” Advances in neural information processing systems, vol. 27, pp. 3104-3112, 2014.
[13] Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555, 2014.
[14] Bai, J. Z. Kolter, V. Koltun “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv:1803.01271, 2018.
[15] Wu, H. , Xu, J. , Wang, L. , and Long, M. “Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting,” Advances in neural information processing systems , pp. 101–112, 2021.
[16] Zhou, H. , Zhang, S. , Peng, J. , Zhang, S. , Li, J. , Xiong, H. , and Zhang, W. , “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in The Thirty Fifth AAAI Conference on Artificial Intelligence, vol. 35, pp. 11106–11115, 2021.
[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Polosukhin, I. “Attention is all you need,” Advances in neural information processing systems, pp. 5998-6008, 2017.
[18] D. Bahdanau, K. Cho, Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv:1409.0475, 2014.
[19] G. E. P. Box and Jenkins G M, “Time series analysis : Forecasting and control,” Journal of Time Series Analysis, vol. 4, no. 2, pp. 45-55, 2019.
[20] N. E. Huang, Z. Shen, S. R. Long, et al, “The empirical mode decomposition of non-stationary signals and the Hilbert spectrum,” Proc. R. Soc. London A, vol. 454, pp. 903–995, 1998.
[21] F. Shen, J. Chao, J. Zhao, “Forecasting exchange rate using deep belief networks and conjugate gradient method,” Neurocomputing, vol. 167, pp. 243-253, 2015.
[22] D. Rumelhart, Hinton, G. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 532–536, 1986.
[23] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," In ICLR, 2015.
[24] G. B. Sahoo, C. Ray, "Flow forecasting for a hawaiian stream using rating curves and neural networks," Journal of Hydrology, vol. 167, no. 1, pp. 243-253, 2015.
[25] Y. Li, H. Wu, and H. Liu, “Multi-step wind speed forecasting using EWT decomposition, LSTM principal computing, RELM subordinate computing and IEW T reconstruction,” Energy Convers. Manage., vol. 167, pp. 203-219, Jul. 2018.
[26] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” Advances in neural information processing systems, pp. 3104-3112, 2014.
[27] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," In ICLR, 2015.
[28] Mohammad Obaidur Rahman, Md. Sabir Hossain, Ta-Seen Junaid, Md. Shafiu Alam Forhad, Muhammad Kamal Hossen, 2019. “Predicting Prices of Stock Market using Gated Recurrent Units (GRUs) Neural Networks,” IJCSNS, vol. 19 no. 1 pp. 213-222, 2019.
[29] S. Ishak and H. Al-Deek, “Performance evaluation of short-term time-series traffic prediction model,” Journal of transportation engineering, vol. 128, no. 6, pp. 490-498, Nov. Dec. 2002.
[30] Z. Liu, P. Jiang, L. Zhang, and X. Niu, “A combined forecasting model for time series: Application to short-term wind speed forecasting,” Applied Energy, vol. 259, Art. no. 114137, Feb. 2020.
[31] G. E. P. Box and Jenkins G M, “Time series analysis : Forecasting and control,” Journal of Forecasting, vol. 3, no. 3, 2010.
[32] E. D. McKenzie, “ General exponential smoothing and the equivalent arma process,” Journal of Forecasting, vol. 3, no. 3, pp. 333-344, 1984.