Journal of computer science and software applications
https://www.mfacademia.org/index.php/jcssa
ISSN:2377-0430
Vol. 5, No. 5, 2025

Multivariate Time Series Forecasting through Automated Feature Extraction and Transformer-Based Modeling

Yu Cheng
Fordham University, New York, USA
ycheng77@fordham.edu

Abstract:

This paper addresses the challenges of feature redundancy and complex high-dimensional dependencies in multivariate time series forecasting. A forecasting method is proposed by combining TSFresh-based feature engineering with the Temporal Fusion Transformer. The method first applies TSFresh to perform automated feature extraction and selection on raw time series data. This process reduces input dimensionality and enhances feature representation. Then, the Temporal Fusion Transformer is used to model temporal dependencies and inter-variable relationships. It integrates dynamic variable selection, gated residual networks, and multi-head attention to achieve accurate future sequence prediction. Experimental results on the Electricity multivariate load dataset show that the proposed model outperforms existing mainstream methods in terms of MAE, MSE, and R^2. It also shows stable performance in hyperparameter sensitivity analysis and robustness testing. These results confirm the effectiveness and reliability of the method in complex multivariate time series forecasting scenarios.

Keywords:

Time Series Forecasting, Feature Engineering, Deep Learning, Transformer

1. Introduction

With the rapid development of information technology, the volume of data generated during system operations has grown significantly. Multidimensional time-related sequence data are especially prevalent in domains such as finance, meteorology, energy, healthcare, and industrial control [1]. These data often exhibit high dynamism, nonlinearity, and complex variable interactions. Such characteristics pose serious challenges for modeling and prediction. Extracting key features from massive, multi-source, and heterogeneous data and building predictive models with strong generalization capabilities has become a crucial topic in time series modeling research [2].

Multivariate time series forecasting requires not only capturing temporal patterns but also understanding the coupling and dynamic dependencies among variables. Traditional statistical models often struggle with high-dimensional data and complex variable interactions. In contrast, deep learning models show strong capabilities in modeling nonlinear relationships, capturing long-term dependencies, and enabling end-to-end prediction. However, the black-box nature of deep models and their strong dependence on input features make their performance highly sensitive to the quality of feature construction. Therefore, discovering latent features from time series data is key to improving model performance.

TSFresh is an automated tool for time series feature extraction. It computes a wide range of statistical, frequency-domain, and structural features from raw sequences and applies statistical tests for feature selection. This process greatly improves both modeling efficiency and feature representation. TSFresh reduces the reliance on manual feature engineering and enhances feature completeness and discriminative

power from multiple perspectives. These improvements provide a solid data foundation for model training and contribute to better predictive accuracy and generalization [3].

On the other hand, the Temporal Fusion Transformer is a novel deep learning architecture designed to address challenges in modeling long-term dependencies and unstable variable relationships in multivariate time series. It integrates self-attention mechanisms, gated recurrent structures, and variable selection modules. The model can capture complex dependencies and multi-scale features. It also includes interpretable components that help analyze the influence of input features. Compared to traditional models, it handles irregular data more flexibly, imputes missing values, and supports multi-step forecasting [4, 5]. As a result, it achieves higher accuracy and robustness across various real-world applications.

In summary, combining TSFresh for feature extraction with the modeling power of the Temporal Fusion Transformer can enhance both the accuracy and efficiency of multivariate time series forecasting. This integrated approach also enables the development of predictive frameworks that are interpretable and robust. Such research not only holds theoretical significance for algorithm development but also offers strong practical value. It supports the deployment and optimization of time series modeling in complex real-world scenarios.

2. Related work

2.1 TSFresh Feature Engineering

Feature engineering plays a critical role in time series modeling tasks. Traditional feature extraction methods often rely on manual design. These methods are subjective and uncertain. They struggle to maintain stable performance in large-scale and high-dimensional data scenarios. To improve the systematization and automation of feature extraction, automated tools for time series feature generation have become a research focus. By transforming raw sequence data and applying statistical mining, these tools can automatically generate representative feature sets. This approach helps address the limitations in expressing multivariate features[6].

TSFresh is a high-dimensional feature extraction framework for time series data. It computes hundreds of statistical, frequency-domain, Fourier, and shift-related features. This allows for a comprehensive representation of data patterns. The method is highly automated and supports various modeling algorithms[7,8]. For feature selection, TSFresh uses a filter-based statistical testing mechanism. It evaluates the significance of correlations between features and target variables. This process reduces redundant dimensions and improves training efficiency and generalization. Compared to manual construction or simple sliding window methods, TSFresh greatly enhances the quality of features and the representativeness of model inputs [9].

Applying TSFresh in multivariate time series modeling strengthens the structural information of input data. It also captures hidden interaction patterns between variables through large-scale feature combinations. This approach is well-suited for complex systems with nonlinearity, nonstationarity, and variable coupling. It provides clean and rich feature inputs for deep prediction models. As a result, TSFresh establishes a solid foundation for end-to-end forecasting systems. It is a key component for achieving high-accuracy time series prediction.

2.1 TSFresh Feature Engineering

The Transformer architecture was originally proposed for sequence modeling tasks. Its design is fully based on the self-attention mechanism[10]. This allows the model to process entire sequences in parallel and capture long-range dependencies. Unlike traditional recurrent mechanisms, it avoids the gradient vanishing

problem in long sequence training. It also improves training efficiency and enhances model representation capacity [11]. In time series modeling, the Transformer has been widely adopted and refined due to its potential in capturing temporal dependencies and complex inter-variable relationships.

Unlike traditional recurrent neural networks, the Transformer can flexibly handle interactions across different time steps. It uses positional encoding to retain the temporal context. For multivariate time series data, this structure is particularly suitable for modeling dynamic dependencies and nonlinear interactions among variables. The self-attention mechanism assigns different attention weights. This enhances learning for key time steps or critical variables. As a result, the model gains interpretability and selectivity. It shows stronger generalization and robustness in complex tasks [12, 13].

In recent years, many improvements to the Transformer architecture have emerged to better suit time series characteristics. These include local attention, causal convolution, and temporal gating mechanisms. Such modifications enhance the model's ability to represent temporal structures [15]. They retain the advantages of the original Transformer in handling high-dimensional sequences. At the same time, they improve performance in long-horizon forecasting, multi-step prediction, and multivariate modeling. The continued evolution of Transformer-based models has become a key direction in deep time series modeling.

3. Method

This paper proposes a multivariate time series forecasting method that integrates feature engineering with deep sequence modeling. The overall framework consists of two key components. First, the TSFresh-based feature engineering module systematically processes the raw multivariate time series. It extracts rich statistical and structural features and applies a significance-based selection mechanism. This produces a high-quality and low-redundancy feature set. Next, the selected features are fed into the Temporal Fusion Transformer model. This model combines interpretable attention mechanisms, dynamic variable selection networks, and gated recurrent structures. It effectively captures complex dependencies across both temporal and variable dimensions. The model enables accurate prediction of future time steps. The architecture is illustrated in Figure 1.

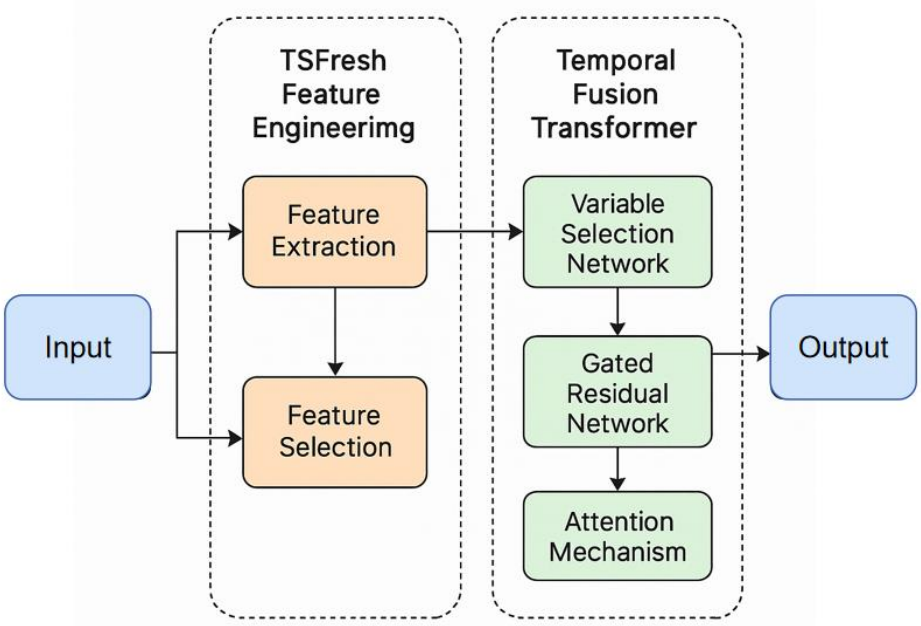

Figure 1. Overall model architecture diagram

3.1 Feature Selection Module

In multivariate time series forecasting, the original input often contains a large number of dimensions and redundant features. Direct input to the deep model will not only increase the computational complexity, but may also cause overfitting and decreased generalization ability. Therefore, compressing and screening the feature space before model training has important theoretical and practical significance. In order to achieve efficient feature selection, this paper adopts a method based on statistical tests and correlation evaluation to perform multi-level processing on the original feature set extracted by TSFresh, eliminate redundant features, and retain subsets that are significantly correlated with the target variable, thereby optimizing the input quality of the downstream model. The model architecture is shown in Figure 2.

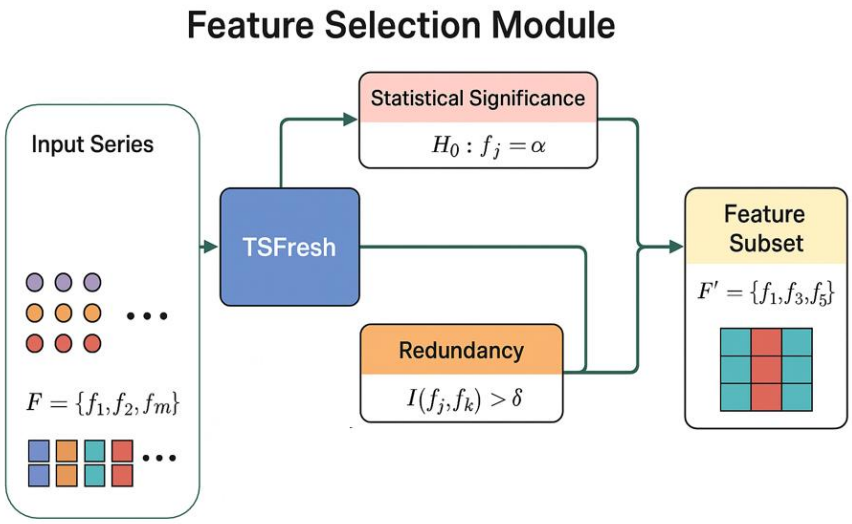

Figure 2. Feature Selection Module Architecture

First, assume that the original multivariate time series data is input as:

$$X = \{x_1^{(t)}, x_2^{(t)}, ..., x_n^{(t)}\}_{t=1}^{T}$$

Where $x_i^{(t)}$ represents the observed value of the i-th variable at time step t, T represents the time length, and n represents the number of variables. After feature extraction through TSFresh, the feature matrix is obtained:

$$F = \{f_1, f_2, ..., f_m\}$$

Where f_j is the jth statistical or frequency domain feature calculated from the original sequence, with a total of m features. Next, the correlation between each feature and the target variable y is evaluated, and its significance is determined using the hypothesis testing method.

For each feature f_j, we set up a null hypothesis H_0: this feature has no significant statistical relationship with the target variable y, and use the p-value as the basis for judgment. If the p-value is less than the set threshold α, then H_0 is rejected and the feature is retained. That is:

$$p(f_j, y) < \alpha \Rightarrow f_j \in F'$$

F' is the set of effective features after screening. Furthermore, in order to reduce the redundancy between features, the mutual information or Pearson correlation coefficient is used to calculate the linear/nonlinear correlation between features, and the redundancy threshold δ is set. If the correlation between two features exceeds δ, the one with higher redundancy is removed.

Finally, the feature selection module outputs a low-dimensional, non-redundant, and highly relevant feature subset F', which serves as the input of the Temporal Fusion Transformer. This module effectively compresses the input dimension while retaining the amount of information, and enhances the stability and predictive ability of the model when facing high-dimensional time series data. Through this pre-processing of data mining, the overall model can achieve end-to-end learning in structure, achieving a better balance between expressiveness and generalization ability.

3.2 Temporal Fusion Transformer Architecture

After completing feature engineering, this paper uses Temporal Fusion Transformer (TFT) as the core prediction model to model and predict the selected multivariate time series features. TFT can simultaneously process static features, historical observations, and future known variables. Combined with multi-level attention mechanisms and gating structures, it has significant advantages in capturing the interaction relationship and time dependency between variables. In order to effectively model the complex dynamic behavior in the input sequence, TFT integrates key components such as learnable position encoding, variable selection network, gated residual connection, and multi-head attention mechanism in structure. Its module architecture is shown in Figure 3.

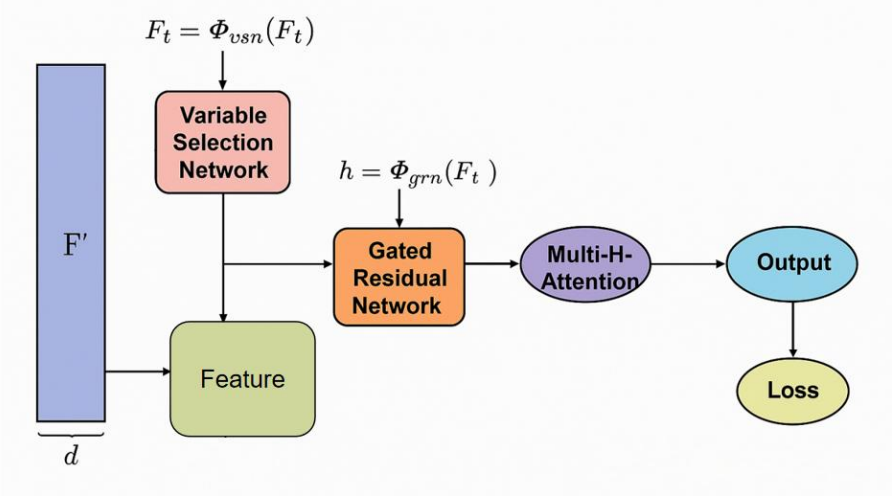

Figure 3. Temporal Fusion Transformer module architecture

Assume that the input after feature selection is the feature matrix $F' \in R^{T \times d}$, where T represents the time step and d represents the number of effective feature dimensions selected. First, a learnable variable selection network ϕ_{vsn} is used to soft-select different feature dimensions to highlight the most important variables at the current moment:

$$F_t = \phi_{vsn}(F') = F' \otimes \sigma(WF' + b)$$

$\sigma(\cdot)$ is the sigmoid activation function, W and b are learnable parameters, and $\otimes$ represents element-by-element multiplication. This structure dynamically controls the weight of the feature dimension input at each time step, allowing the model to focus on key variables.

Subsequently, the features processed by variable selection are fed into the gated residual network ϕ_{grn} to model nonlinear time dependency while maintaining information stability and gradient propagation capability:

$$h_t = \phi_{grn}(F_t) = GLU(V \cdot RELU(UF_t) + b') + F_t$$

U, V, b' is a trainable parameter, GLU represents a gated linear unit, ReLU is an activation function, and the residual connection structure enhances the network's ability to capture deep information. In the time dimension, the representations of multiple time steps will be fed into the multi-head self-attention module to model the global time-dependent pattern and generate a fused context feature representation C_t.

Finally, after obtaining the predicted output y_t, the model minimizes the error between the predicted value and the true value through the supervised learning objective, and the loss function used is the weighted mean square error (Weighted MSE):

$$L = \frac{1}{T}\sum_{t=1}^{T} w_t (y_t - y'_t)^2$$

Where w_t is the loss weight factor of each time step, which is used to control the contribution of different time points to the overall loss. This loss function not only considers the global minimization of the prediction error, but also optimizes the accuracy of the key time period through the time weighting mechanism, so that the model can still maintain robust performance under multi-step prediction and unbalanced sample distribution.

4. Experimental Results

4.1 Dataset

This study uses the Electricity Load Diagrams 20112014 (Electricity) dataset as the benchmark for multivariate time series forecasting. The dataset contains electricity load data from 370 clients in a specific region, collected between 2011 and 2014. The data are recorded every 15 minutes. It includes households, industrial users, and commercial entities. The series shows clear periodicity, trends, and fluctuations. This makes the dataset suitable for tasks such as load forecasting and energy consumption modeling.

In the experimental setup, a subset of user load data is selected as input variables. The load data of a designated target user are used as the prediction target. This forms a multi-input single-output forecasting task. To improve modeling efficiency and reproducibility, the data are standardized. Then, based on a fixed sliding window strategy, the data are split into training, validation, and test sets. Each sample consists of a historical observation sequence and the corresponding future prediction target.

The dataset has several advantages. It is continuous, well-structured, and strongly periodic. There are also correlations among different variables. These properties provide a solid foundation for evaluating the

proposed model in multivariate modeling and long-term forecasting tasks. Moreover, the high-frequency sampling and long duration allow for comprehensive assessment of model stability and robustness under varying time scales, forecast horizons, and input dimensions.

4.2 Experimental Results

In this section, this paper first gives the comparative experimental results of the proposed algorithm and other algorithms, as shown in Table 1.

Table 1: Comparative experimental results

Method	**MAE**	**MSE**	**R^2**
Ours	0.184	0.256	0.861
Informer[15]	0.211	0.283	0.844
Autoformer[16]	0.198	0.270	0.850
LSTM + Attention[17]	0.237	0.305	0.832
Temporal Convolutional Network (TCN)[18]	0.249	0.319	0.818

As shown in Table 1, the proposed method achieves the best overall performance in the multivariate time series forecasting task. Specifically, it outperforms other advanced models on major evaluation metrics such as MAE, MSE, and R^2. This demonstrates strong modeling capability and high prediction accuracy. The MAE is 0.184, and the MSE is 0.256. These results indicate that the model maintains stable error control and effectively reduces both point-wise and overall bias.

In comparison, although Informer and Autoformer also belong to the Transformer family and are capable of modeling long sequences, their performance on error metrics is slightly inferior. Informer achieves an MAE of 0.211, an MSE of 0.283, and an R^2 of 0.844. Autoformer performs slightly better than Informer in error control but still falls short of the proposed method. This suggests some limitations in feature selection or variable modeling in these models.

The LSTM + Attention model, a classical deep sequence architecture, records an MAE of 0.237, an MSE of 0.305, and an R^2 of 0.832 in this experiment. Its overall performance is clearly weaker than that of Transformer-based models. This result shows that traditional recurrent structures have difficulty capturing long-term dependencies in high-dimensional and complex interactive data. Problems such as information loss and gradient decay are more likely to occur.

The TCN model shows the weakest predictive performance. Its R^2 is only 0.818, indicating a limitation in global sequence modeling. Among all models, it has the highest error values. This suggests that its representational capacity is limited when handling highly nonlinear and strongly coupled multivariate time series. In summary, the proposed method combines feature engineering with the self-attention mechanism. It can more effectively capture key variable information and temporal dependencies, leading to better forecasting results.

Furthermore, this paper presents a hyperparameter sensitivity experiment. First, the experimental results of the learning rate are given, as shown in Table 2.

Table 2: Hyperparameter sensitivity experiments (Lr)

Lr	MAE	MSE	R^2
0.005	0.211	0.289	0.842
0.004	0.201	0.274	0.849
0.003	0.192	0.262	0.855
0.002	0.188	0.258	0.858
0.001	0.184	0.256	0.861

As shown in Table 2, the learning rate (Lr) has a significant impact on model performance. As the learning rate gradually decreases, the model shows continuous improvement across all evaluation metrics. Specifically, both MAE and MSE values decrease as the learning rate becomes smaller, indicating better control of prediction errors. At the same time, the R^2 metric shows an upward trend, suggesting enhanced ability to explain data variance and improved overall fitting performance.

When the learning rate is relatively high, for example Lr = 0.005, the model records an MAE of 0.211, an MSE of 0.289, and an R^2 of 0.842. The predictive performance at this stage is relatively poor. This is mainly because a high learning rate often causes oscillations during training, making it difficult for the model to converge to an optimal solution. As the learning rate decreases to 0.001, the MAE drops to 0.184, the MSE falls to 0.256, and the R^2 rises to 0.861. The model becomes more stable and better captures complex patterns and dynamic relationships in the time series.

Overall, reasonably reducing the learning rate can effectively enhance model performance. However, an excessively low learning rate may lead to prolonged training time or getting stuck in local optima. Based on the experimental results, Lr = 0.001 achieves the best balance. It maintains low errors while ensuring good convergence speed and predictive capability. Therefore, Lr = 0.001 will be used as the default learning rate for further validation and testing in subsequent experiments.

Next, the optimizer analysis of the hyperparameter sensitivity experimental results is given, and the experimental results are shown in Table 3.

Table 3: Hyperparameter sensitivity experiments(Optimizer)

Optimizer	MAE	MSE	R^2
SGD	0.236	0.302	0.834
AdaGrad	0.221	0.284	0.843
Adam	0.196	0.264	0.854
AdamW	0.184	0.256	0.861

As shown in Table 2, different optimizers have a significant impact on the training performance of the model. The overall trend indicates that as optimizer performance improves, the model shows better results in terms of MAE, MSE, and R^2. The traditional SGD optimizer performs relatively poorly in this task, with an MAE of 0.236, MSE of 0.302, and R^2 of only 0.834. This suggests that SGD converges slowly and is prone to getting stuck in local minima when dealing with complex multivariate time series data.

In comparison, AdaGrad improves model performance to some extent, particularly in reducing MSE. However, due to its gradient accumulation effect, the learning rate becomes too small in later stages, resulting in suboptimal final fitting. The Adam optimizer introduces an adaptive learning rate mechanism. It significantly reduces error metrics, with the MAE dropping to 0.196, the MSE to 0.264, and the R^2 increasing to 0.854. The training process becomes more stable and efficient.

Overall, the model achieves the best predictive performance when using the AdamW optimizer. Compared with other optimizers, AdamW introduces an improved weight decay strategy that effectively reduces overfitting. It helps the model maintain low error while further enhancing generalization. Under AdamW, the model reaches an MAE of 0.184, an MSE of 0.256, and an R^2 of 0.861. This indicates that AdamW is better suited for complex time series forecasting tasks. Therefore, AdamW is used as the default optimizer in subsequent experiments.

Furthermore, this paper also gives the analysis results of the impact of the number of input variables on the model stability, as shown in Figure 4.

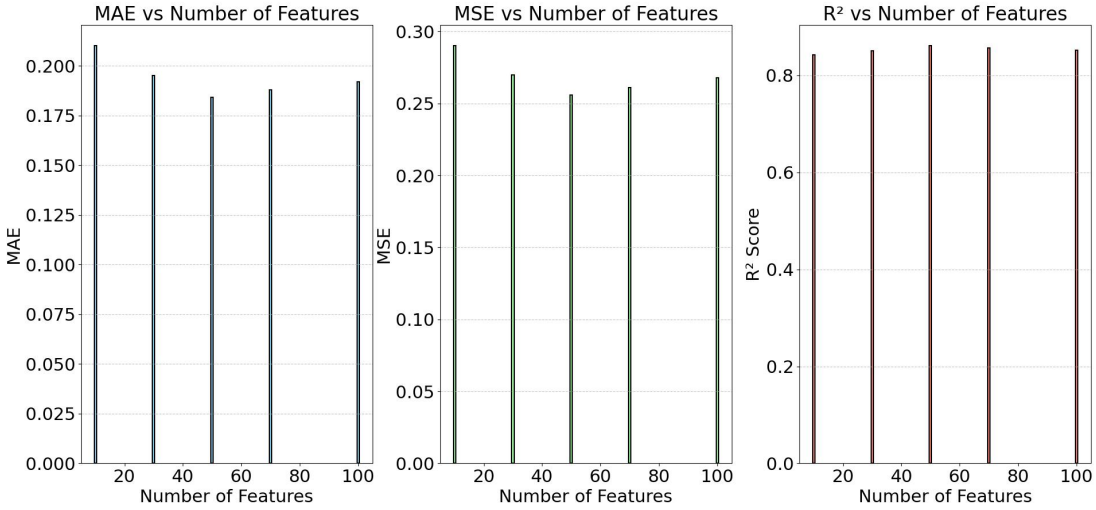

Figure 4. Results of analysis on the impact of the number of input variables on model stability

As shown in Figure 4, the model's performance on the three main evaluation metrics changes with the number of input variables. When the input dimension is small (e.g., 10 features), the model shows higher errors. Both MAE and MSE are relatively large. This indicates that limited information restricts the model's ability to learn complex temporal dependencies. As the number of features increases from 10 to 50, the MAE and MSE gradually decrease, and the R^2 value rises. This suggests that a moderate number of input variables helps the model better capture system structure.

When the number of features reaches 50, the model achieves its best performance. The MAE is as low as 0.184, the MSE is 0.256, and the R^2 reaches 0.861. This result indicates that, at this level, the model can fully learn variable interactions and capture dynamic changes in the sequence. Accurate prediction is achieved. These findings also confirm the importance of feature selection strategies. Proper selection and organization of input features can improve both model stability and generalization.

However, when the input dimension continues to increase to 70 or even 100, the model performance becomes slightly unstable. Error metrics rise slightly, and R^2 decreases. This may be caused by redundant features that increase model complexity. Such redundancy can lead to overfitting or difficulties in gradient

propagation. Therefore, the choice of input variables should balance between providing enough information and avoiding excessive redundancy. This balance is key to achieving optimal stability and accuracy.

Finally, this paper also gives a robustness test experiment under noisy data, and the experimental results are shown in Figure 5.

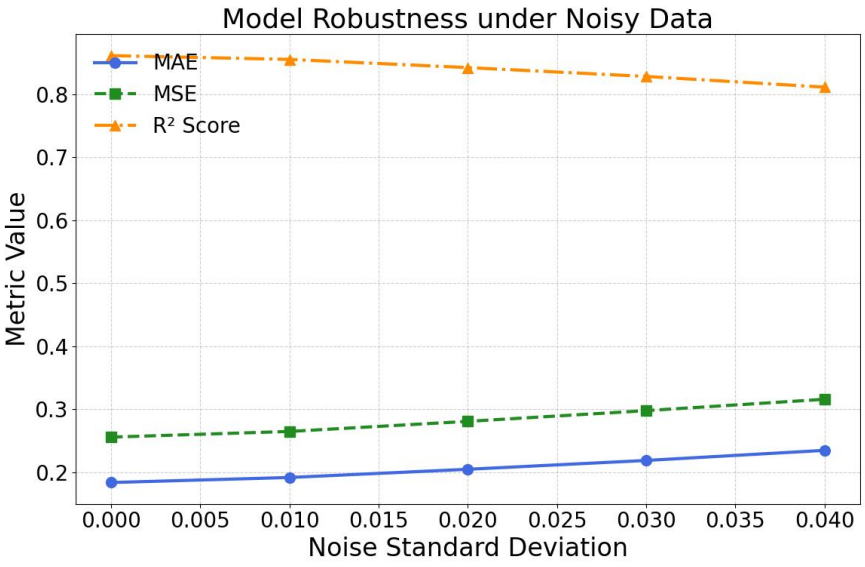

Figure 5. Robustness test experiment under noisy data

As shown in Figure 5, the model's performance gradually decreases as the noise level increases. This indicates that noise interference affects model stability. Specifically, both MAE and MSE increase with the rise of noise standard deviation. The MAE grows from 0.184 (no noise) to 0.235 under the highest noise level. The MSE rises from 0.256 to 0.316. These results suggest that prediction errors become larger as noise intensifies.

At the same time, the R^2 value drops steadily from 0.861 to 0.811. This shows that the model's ability to explain data variance weakens under noise. The downward trend also confirms the model's sensitivity to data perturbations. In long-term forecasting or tasks involving strong high-dimensional variable interactions, external noise may further propagate prediction errors.

Although performance decreases, the overall change remains stable. The model still maintains strong generalization and robustness. Even when the noise standard deviation reaches 0.04, the R^2 stays above 0.8. Both MAE and MSE remain within a reasonable range. This demonstrates that the model has a certain level of noise resistance and can retain stable prediction performance in noisy environments.

5. Conclusion

This study proposes a deep learning model for multivariate time series forecasting by integrating TSFresh-based feature engineering with the Temporal Fusion Transformer. The method automatically extracts statistical and structural features from high-dimensional time series data. It combines dynamic variable selection with multi-layer attention mechanisms. This enhances the model's capability to learn in complex data environments and improves prediction accuracy. Experimental results show that the proposed method outperforms mainstream models on several key metrics. It demonstrates strong generalization and robustness, especially under high input dimensionality or noise interference. Comparative experiments and

hyperparameter sensitivity analyses validate the importance of feature selection and structural optimization. Factors such as learning rate, optimizer, and the number of input features significantly influence model performance. These findings highlight the necessity of constructing high-quality input features and applying well-designed training strategies. Moreover, robustness tests under noisy data conditions show that the model can effectively handle uncertainty and data disturbances in practical applications. This confirms its practical value and potential for engineering deployment.

The proposed approach is not only innovative in model design but also offers a new reference for time series forecasting in real-world tasks such as electricity load forecasting, traffic flow prediction, and financial trend modeling. In particular, when dealing with multi-source heterogeneous data and high-dimensional dynamic variables, the framework supports effective feature compression, information filtering, and efficient prediction. It provides a scalable and interpretable solution for sequence modeling in complex systems. Future research can further explore the model's potential in multi-task forecasting, multimodal data fusion, and graph-based sequence modeling. In the realm of industrial applications, work can also concentrate on automated feature engineering, model compression, and real-time forecasting to address the demands of higher frequency, larger scale, and greater timeliness. These directions can facilitate the deployment of time series analysis in intelligent decision-making and data-driven systems.

References

[1] Verdonck, Tim, et al. "Special issue on feature engineering editorial." Machine learning 113.7 (2024): 3917-3928.

[2] Morid, Mohammad Amin, Olivia R. Liu Sheng, and Joseph Dunbar. "Time series prediction using deep learning methods in healthcare." ACM Transactions on Management Information Systems 14.1 (2023): 1-29.

[3] Wang, Min, Hua Wang, and Fan Zhang. "Famc-net: Frequency domain parity correction attention and multi-scale dilated convolution for time series forecasting." Proceedings of the 32nd ACM international conference on information and knowledge management. 2023.

[4] Ramesh, Ganapathy, et al. "Prediction of energy production level in large PV plants through AUTO-encoder based neural-network (AUTO-NN) with restricted Boltzmann feature extraction." Future Internet 15.2 (2023): 46.

[5] Liang, Yuxuan, et al. "Foundation models for time series analysis: A tutorial and survey." Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining. 2024.

[6] Yusuf, Shefiu, Sadis Bello, and James Brown. "AI-Driven Feature Engineering: Tools and Best Practice." (2023).

[7] Zhang, Wei, and Yu Dai. "A multiscale electricity theft detection model based on feature engineering." Big Data Research 36 (2024): 100457.

[8] John, Beauden. "Automated Feature Engineering: Tools, Techniques, and Future Prospects." (2025).

[9] Liu, Huijie, Qingsheng Zhao, and Dingkang Liang. "Electricity theft detection model based on feature engineering and integrated classifier." 2024 8th International Conference on Electrical, Mechanical and Computer Engineering (ICEMCE). IEEE, 2024.

[10] Zeng, Ailing, et al. "Are transformers effective for time series forecasting?." Proceedings of the AAAI conference on artificial intelligence. Vol. 37. No. 9. 2023.

[11] Woo, Gerald, et al. "Unified training of universal time series forecasting transformers." (2024): 53140.

[12] Liu, Yong, et al. "itransformer: Inverted transformers are effective for time series forecasting." arXiv preprint arXiv:2310.06625 (2023).

[13] Zhang, Yunhao, and Junchi Yan. "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting." The eleventh international conference on learning representations. 2023.

[14] Lezmi, Edmond, and Jiali Xu. "Time series forecasting with transformer models and application to asset management." Available at SSRN 4375798 (2023).

[15] Zhou, Haoyi, et al. "Informer: Beyond efficient transformer for long sequence time-series forecasting." Proceedings of the AAAI conference on artificial intelligence. Vol. 35. No. 12. 2021.

[16] Wu, Haixu, et al. "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting." Advances in neural information processing systems 34 (2021): 22419-22430.

[17] Abbasimehr, Hossein, and Reza Paki. "Improving time series forecasting using LSTM and attention models." Journal of Ambient Intelligence and Humanized Computing 13.1 (2022): 673-691.

[18] Ehteram, Mohammad, and Elham Ghanbari-Adivi. "Self-attention (SA) temporal convolutional network (SATCN)-long short-term memory neural network (SATCN-LSTM): an advanced python code for predicting groundwater level." Environmental Science and Pollution Research 30.40 (2023): 92903-92921.