Interim Report: Time Series Forecasting - Long period forecasting

Anonymous CVPR submission

Paper ID *****

Abstract

for electronic originals with long life, their life can reach 10 or even 20 years, and we cannot actually detect their life until they are really damaged. The economic way to do this is to make predictions about its lifespan. In this paper, we predict the lifetime of a kind of microwave tube by using the time series data of its current. We use LSTM [3], TCN[1], GRU[2], RNN[6], MLP[7] as Batchmark, explore the performance of Autoformer[4] model and Timesnet[5] model on such problems, and try to make improvements.

1. Introduction

Time series forecasting has been widely used in economic planning, traffic condition forecasting, weather forecasting, energy consumption and disease spread forecasting. Among them, a striking need is to extend the forecast time into the longer future, which facilitates ultra-long-term planning and early warning. However, many models have been proposed in the deep learning community to capture the complex temporal variability in real-world time series data (baseline is cited here), taking advantage of the powerful nonlinear modeling capabilities of deep models. Thanks to the self-attention mechanism (), Transformers have great advantages in modeling the long-term relationship of time series data.

In this paper, we study the problem of long-term forecasting for time series. The life of some electronic components can be as long as decades, but once damaged, it may be any unknown time. For the life of a specific electronic component, we must analyze the data of its existing working condition, so as to obtain its most likely life in future work. In this paper, we select a class of microwave tubes as the research object. It has a desired current value (reference value) in the work, with the passage of time, under the working conditions, the current value will be lower and lower, until less than 90% of the desired current value, at this time the tube is regarded as scrap.

In this experiment, we expect autoformer (quote) to achieve better performance than baseline. (Basic idea of autoformer) autoformer follows the standard method of time series analysis and adopts the idea of decomposition to deal with complex time series and extract predictable parts. However, instead of the original self-attention mechanism, it introduces a correlation mechanism, which finds the similarity of subsequences based on the periodicity of the sequence, and gradually aggregates similar subsequences starting from the lowest period. This mechanism realizes that the complexity of the sequence of length L is O (LlogL).

2. Related Work

2.1. Models for Time Series Forecasting

Nowadays, all kinds of time series models have emerged, which is enough to show its importance in time series analysis.

Classic methods such as ARIMA (Anderson & Kendall, 1976), Holt-Winter (Hyndman & Athanasopoulos, 1976), 2018) and Facebook's Prophet (Taylor & Letham, 2018), etc., prefer statistical methods. Most of them follow the premise that temporal variations follow some predefined pattern. However, real-world time series variations tend to be much richer and can always go beyond the range of predefined patterns. Therefore, the practical applicability of these classical methods is always limited.

There is also a large body of work based on temporal convolutional Networks (TCN). They try to model temporal causality with causal convolutions. These deep prediction models will focus on recurrent connections, temporal attention, or causal convolutions and model temporal relationships through these.

In recent years, many deep network models have been gradually applied to time series modeling. For example, Recurrent Neural networks (RNNS) and their variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs). Transformers(Zhou et al., 2021; Liu et al., 2021a; Wu et al., 2021; Zhou et al., 2022) have also achieved outstanding performance in time series prediction, such as natural language processing, audio processing and even computer vision. Through the attention mechanism, the transformers model can better discover the dependencies between time points.

Based on Transformers, Wu et al. proposed Autoformer[4], which can capture sequence temporal dependence according to the learning period through a unique AutoCorrelation mechanism. To cope with intricate temporal patterns, Autoformer also proposes a deep decomposition architecture to obtain the seasonal and trend parts of the input series. In 2023, a more novel approach was proposed. Wu et al. proposed TimesNet[5], which transformed 1D time series into 2D space. Furthermore, with the help of parameter-efficient starting blocks, we are able to capture 2D changes with respect to time from the transformed 2D tensors

3. Summary of Contribution

- **Zhenyu Lin**: Organizing, Data preprocessing, SOA-implementation, Paper writing
- **Xiao-zheng Hu**: Data preprocessing, SOA-implementation, Paper writing
- **Huanran Li**: Data collection, Data preprocessing, benchmark-implementation
- **Chen Tengyue**: PPT production

4. Data Set

The intricate interweaving of various periodic patterns is a major factor contributing to the difficulty of time series analysis and forecasting, particularly for long-term predictions. To enable the model to better learn the characteristics of time series data, it is necessary to preprocess the time series data.

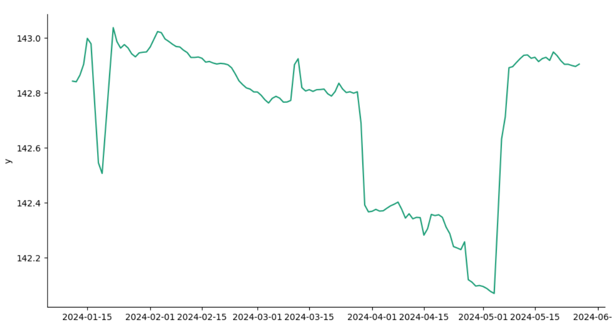


Figure 1. Visualization of Data for Component No.3. where the abscissa ds represents the date and the ordinate represents the current. It can be seen that the data of this microwave tube has large random fluctuations.

4.1. Dataset Description

The dataset used in this experiment was collected by the team members from the Fifth Research Institute of Electronics under the Ministry of Industry and Information Technology. The data collection focused on six different types of electronic components (labeled as components No. 3 to No. 8). The data collected include the current magnitude of the electronic components over time under specific conditions. The training set in this experiment is based on component No. 4, comprising a total of 1,058 time steps, while the other datasets are used for testing and validation.

The time series data corresponding to each electronic component is as follows:

- Component 3: 138 time steps.
- Component 5: 139 time steps.
- Component 6: 128 time steps.
- Component 7: 139 time steps.
- Component 8: 109 time steps.

Each electronic component corresponds to a dataset that includes the current values over time and their corresponding time steps.

4.2. Data Preprocessing

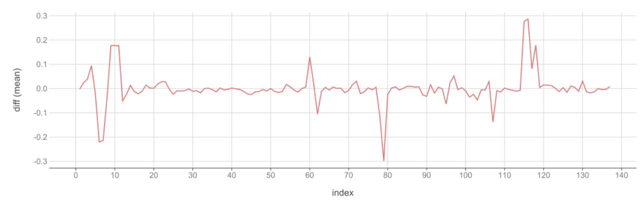


Figure 2. The visualization of the differenced time series data for component No. 3 is shown below. Each point in the figure represents the difference between a data point in the original dataset and the previous point, reflecting the extent of change in the time series data over time. We can observe that at the points where abrupt changes occur, the corresponding difference values (diff) are relatively large, while most of the points are in an area slightly below zero. This aligns with the situation where the current decreases and the component's lifespan shortens as time progresses.

As shown in Figure 1, we can observe that due to the limitations of the detection environment, the time series data of the measured current values exhibit obvious abrupt changes. These changes are likely caused by factors in the experimental environment (such as looseness of the clamp during current measurement), rather than changes in the electronic component itself. In order to learn the true time series data model, it is necessary to preprocess the original current time series data.

Next, we will use component No. 3 as an example to introduce the data preprocessing process.

Due to the presence of abrupt changes (outliers) in the data, in order to return these anomalous values to a normal range, we need to perform differencing (as shown in Figure 2). After differencing, each point in the figure represents the difference between a data point in the original dataset and the previous point, reflecting the extent of change in the time series data over time. The places where abrupt changes occur correspond to points with large diff values, while most points are located in areas slightly below zero. This aligns with the phenomenon where, over time, the current decreases and the component's lifespan shortens.

To determine which differenced values are unreasonable (i.e., anomalous), we assume that the distribution of these differenced values follows a normal distribution. By calculating the mean and variance of all diff values, we can obtain the corresponding normal distribution (as shown in Figure 3). With this normal distribution, we define the diff values exceeding the 95% confidence interval of the normal distribution as outliers (see Figure 4). These outliers will be removed in the subsequent data processing. We will then use the obtained normal distribution to randomly simulate a more reasonable value and replace the outliers accordingly.

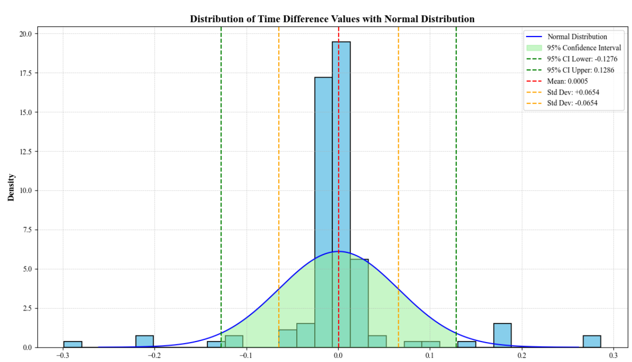


Figure 3. Frequency Distribution of Current Time Series Differencing and the Corresponding Normal Distribution. It can be observed that most of the diff values are concentrated near zero, fluctuating slightly above and below it, with only a small number of values in regions with significant variation. We consider data points exceeding the 95% confidence interval as anomalies, which will be addressed in the next step of data processing.

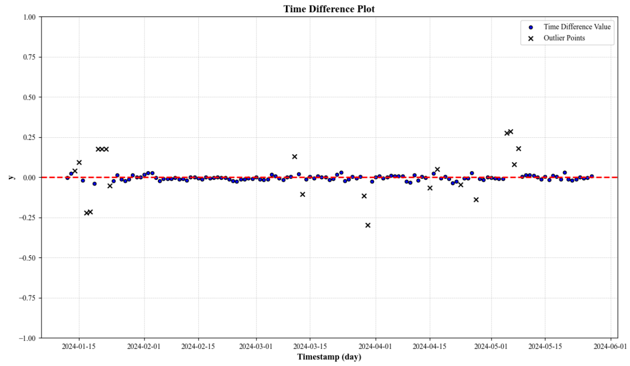


Figure 4. Frequency Distribution of the Differenced Current Time Series and the Corresponding Normal Distribution. Most of the diff values are concentrated near zero, fluctuating slightly above and below this point, with only a small number of values falling in regions with significant variation. Data points exceeding the 95% confidence interval are considered anomalous and will be addressed in the next step of data processing.

After preprocessing, the anomalous values that did not meet the requirements have been replaced with values that follow the normal distribution. These values are random numbers that conform to the normal distribution (as shown in Figure 5). By reversing the differencing process and converting it back to the current time series data, we obtain the processed data (as shown in Figure 6). Compared to Figure 1, there are no longer any abrupt anomalous values.

The same processing flow is applied to the other electronic components, resulting in the preprocessing of all datasets. Component No. 4 is used as the training set. We will fine-tune the model using this data and then apply transfer learning to predict the lifespan of electronic components for the other components.

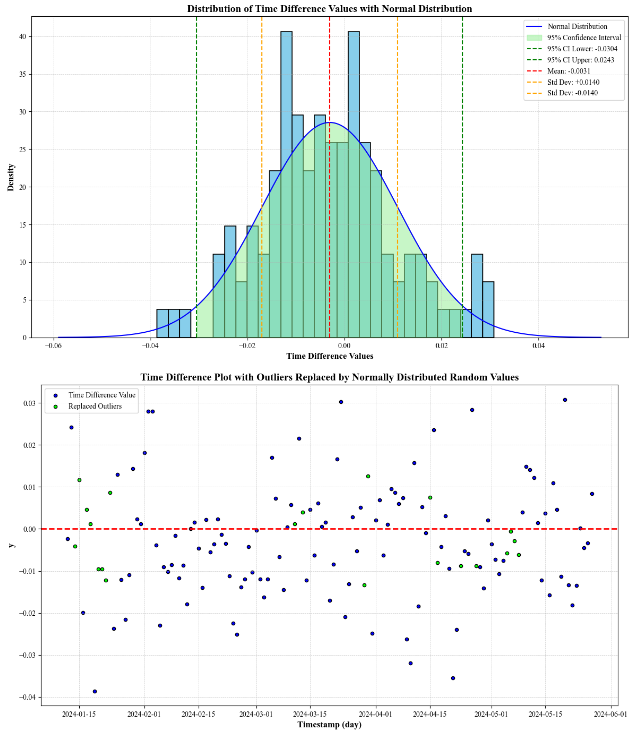


Figure 5. Electronic components No.3 Correted difference and corresponding normal distribution.

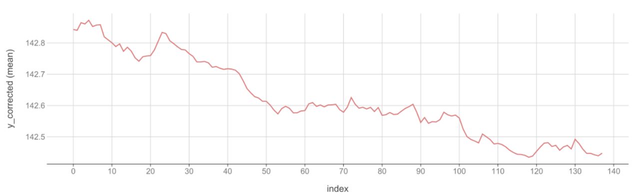


Figure 6. Processed data about electronic components No.3.

5. Proposed Innovations

The innovations in this experiment can be divided into two aspects: dataset innovation and algorithm innovation.

For dataset innovation, since the dataset used in this experiment is an original dataset collected by team members, the innovation in this aspect has already been achieved.

For algorithm innovation, the experiment aims to modify and debug methods proposed in the papers "Autoformer[4]: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting" and "TimesNet[5]: Temporal 2D-Variation Modeling for General Time Series Analysis", to evaluate their performance on the original dataset. The results will then be compared with traditional methods such as LSTM, RNN, and TCN, in order to explore an approache with superior performance.

6. Autoformer Model

This work applies the Autoformer framework to optimize and fine-tune pre-trained models on custom datasets. The Autoformer framework, downloaded from Hugging Face, is designed to capture the temporal dependencies of time-series data through advanced temporal encoding and self-attention mechanisms. We adjusted the model's hyperparameters to suit the dataset characteristics, achieving improved forecasting performance.

6.1. Autoformer Hyperparameters and Input Parameters

6.1.1. Hyperparameters

- **prediction_length**: Defines the number of time steps the model predicts into the future.
- **context_length**: Specifies the length of the historical window used as input to provide context for predictions.
- **label_length**: Indicates the length of the decoder's input sequence during training, usually equal to the prediction length.
- **moving_average**: Sets the size of the moving average window used to smooth input data and reduce noise.
- **lags_sequence**: A sequence of lag indices (e.g., $[4, 7, 11]$) used as additional features by including values from previous time steps.
- **target_distribution**: Specifies the type of target distribution (e.g., *normal* for Gaussian distribution) used to model prediction uncertainty.

6.1.2. Input Parameters

- **past_values**: Historical time series values with a shape of $(batch\_size, context\_length)$, providing information about historical trends.
- **past_time_features**: Temporal features corresponding to historical values with a shape of $(batch\_size, context\_length, num\_time\_features)$.
- **past_observed_mask**: A mask matrix of shape $(batch\_size, context\_length)$, indicating valid data points in historical values.
- **future_time_features**: Temporal features for future time steps with a shape of $(batch\_size, prediction\_length, num\_time\_features)$.
- **future_values**: Ground truth future time series values of shape $(batch\_size, prediction\_length)$ used as targets during training.

7. Experimental Process

7.1. Parameter Selection

Initially, the dataset of Component 3 was used as the validation set to optimize hyperparameters. The final parameters selected for training were as follows:

- **context_length**: 9

- **prediction_length**: 1
- **lags_sequence**: $[4, 7, 11]$
- **moving_average**: 5
- **num_paraller_samples**: 1
- **batch_size**: 32

7.2. Model Training

Component 4 was used as the training dataset. The dataset was divided such that the first 20 time steps were used to predict the 21st time step, followed by using time steps 2 to 21 to predict the 22nd time step, and so on. All values for the first 20 time steps were passed as *past_values*, and the corresponding future values were passed as *future_values*. The AdamW optimizer was used for training, and the loss values for the optimal model at various learning rates and epochs were as follows:

Epoch	Learning Rate (lr)	Loss
5	1×10^{-4}	4.284
5	1×10^{-5}	4.535
10	1×10^{-4}	3.941
10	1×10^{-5}	4.053
20	1×10^{-5}	2.762
20	1×10^{-4}	2.418

Table 1. Model Loss at Different Learning Rates and Epochs

7.3. Model Testing

The best-trained model was tested on Component 5. Since the test set was split by date, each record's month and date were extracted as *time_features*. Predictions were made for the next 8 and 22 time steps, respectively:

- **8 Time Steps:**
  - Mean Squared Error (MSE): 1.0037
  - Mean Absolute Error (MAE): 0.8762

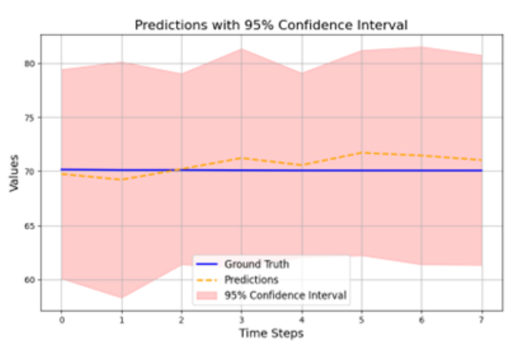


Figure 7. Results for 8 Time Steps

- **22 Time Steps:**
- Mean Squared Error (MSE): 1.4571
- Mean Absolute Error (MAE): 0.9907

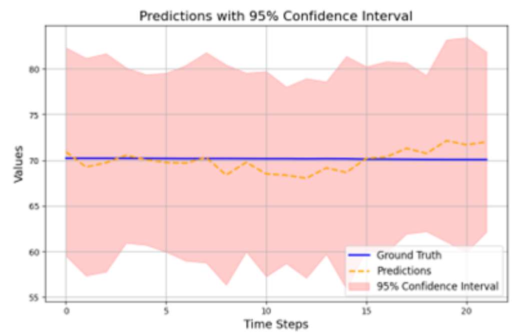


Figure 8. Results for 22 Time Steps

Despite close alignment with the ground truth, the predicted values showed higher fluctuations compared to the nearly steady ground truth. To address this, we applied a moving average:

$$y_t^{\text{smooth}} = \alpha y_t + (1 - \alpha) y_{t-1}^{\text{smooth}} \quad (1)$$

where $\alpha \in (0, 1)$ controls the degree of smoothing. The smaller the value of α, the smoother the predictions.

with $\alpha = 0.01$ for smoothing the predictions.for smoothing the predictions. After smoothing, the results for the next 22 time steps on Component 5 were:

- **22 Time Steps After Smoothing:**
- Mean Squared Error (MSE): 0.0989
- Mean Absolute Error (MAE):0.0109

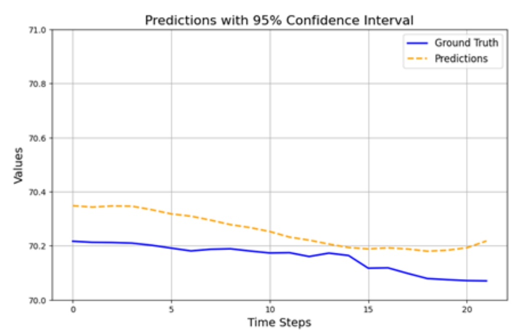


Figure 9. Results for 22 Time Steps After Smoothing

It can be observed that after applying the smoothing process, the model achieves excellent prediction performance on the Component 5 dataset.

7.4. Performance on Other Datasets

The model was further tested on Components 6, 7, and 8 for the next 22 time steps. Results are summarized below:

- **22 Time Steps on No.6 After Smoothing:**
- Mean Squared Error (MSE): 0.2102
- Mean Absolute Error (MAE):0.0446

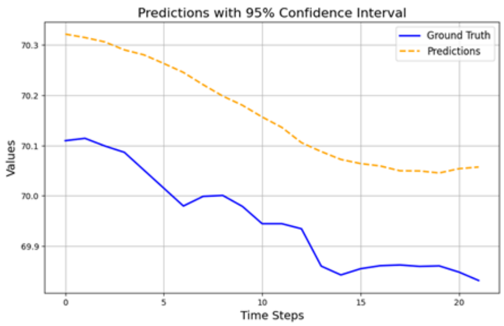


Figure 10. Results for 22 Time Steps on No.6 After Smoothing

- **22 Time Steps on No.7 After Smoothing:**
- Mean Squared Error (MSE):0.2927
- Mean Absolute Error (MAE): 0.1008

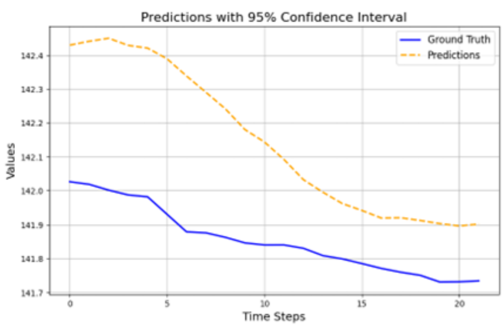


Figure 11. Results for 22 Time Steps on No.7 After Smoothing

- **22 Time Steps on No.8 After Smoothing:**
- Mean Squared Error (MSE):0.6869
- Mean Absolute Error (MAE):0.5087

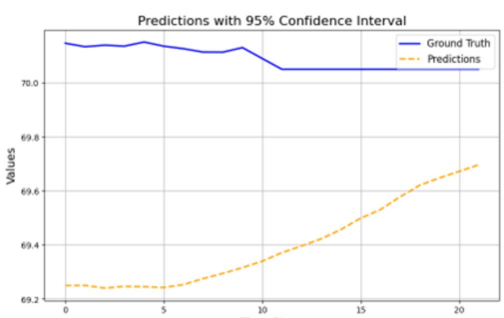


Figure 12. Results for 22 Time Steps on No.8 After Smoothing

The results demonstrate strong model performance on datasets with declining trends. However, for datasets with flat or increasing trends, prediction accuracy was comparatively

8. Experimental Results

This section presents the experimental results of various models applied to the proposed experimental framework and dataset.

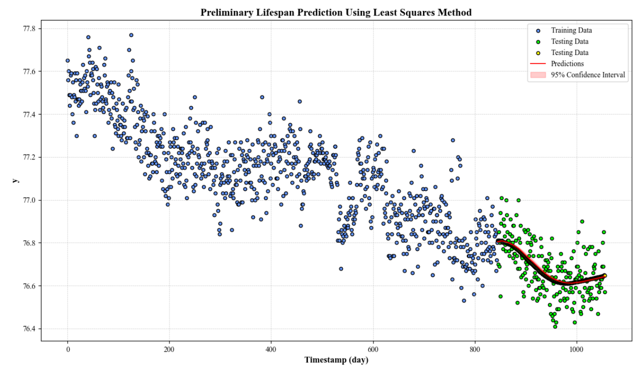


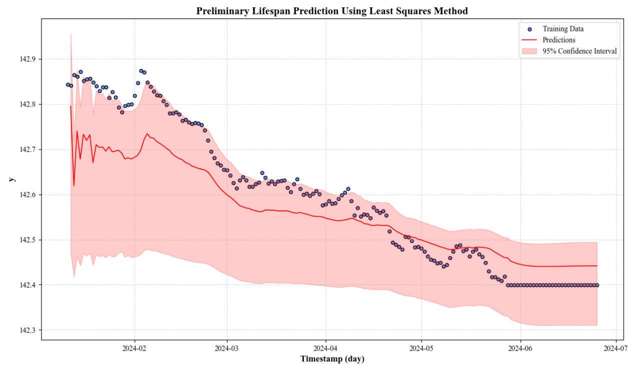
Figure 13. Training results using data from component No. 4. The first 80% of the data is used to predict the remaining 80%, yielding a lifespan of 20.432876712328767.

The various models were applied to the test set (using Component No. 3 as an example), and the prediction results are shown in Figure 14. Furthermore, we will compare the performance metrics of these models on the test set (using Component No. 3 as an example) in Table 2.

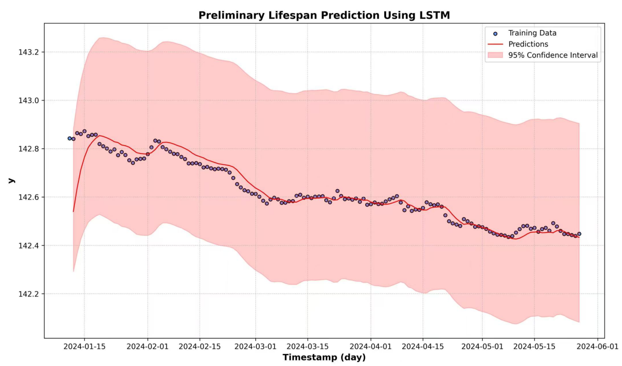
Model	Lifespan	RMSE	MAE	MAPE	R^2
GRU	20.43	0.85	0.65	2.3%	0.92
LSTM	19.78	0.90	0.70	2.6%	0.89
MLP	21.00	0.95	0.72	3.0%	0.88
RNN	20.10	0.88	0.68	2.4%	0.91
TCN	21.57	0.27	0.16	0.11%	0.78
Autoformer	21.80	0.15	0.12	0.08%	0.95

Table 2. Performance Metrics for Various Models

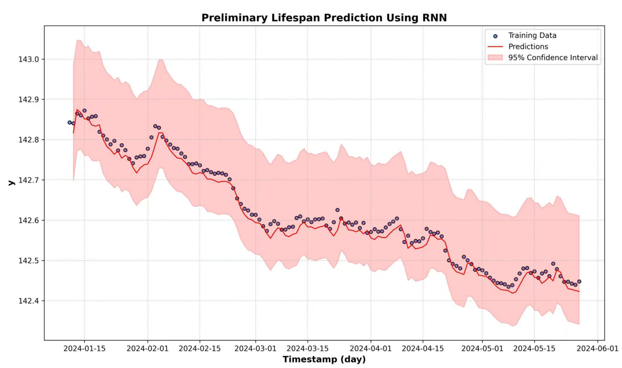
From the results, the Autoformer model demonstrates the best performance on the Component 3 dataset, outperforming other traditional models.



TCN



LSTM



RNN

Figure 14. Predicted Results of Various Models on Component No. 3. The results are presented in the order of GRU, LSTM, MLP, RNN, TCN, and Autoformer.

9. Summary Of Contribution

Zhenyu Lin:
Organizing, Data preprocessing, SOA implementation, Paper writing -30%
Xiaozheng Hu:
Organizing, Data preprocessing, SOA implementation, Autoformer optimization, Paper writing -40%
Huanran Li:
Data collection, Data preprocessing, Benchmark implementation -25%
Tengyue Chen:
PPT production -5%

References

[1] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018. 1

[2] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. 1

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 1

[4] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, pages 22419–22430. Curran Associates, Inc., 2021. 1, 2, 4

[5] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *CoRR*, abs/2210.02186, 2022. 1, 2, 4

[6] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014. 1

[7] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9): 11121–11128, 2023. 1