SHAP for Visual Explanations of Time Series Models

COMP 5331 Knowledge Discovery in Databases
Group 18 Implementation Project: Final Report

23 November 2020

Members

The following are details and information on the members of Group 18:

- LIU Sin Tai (20428105)
  - Supervisor: Prof. XIA Lucy
  - Research Topic: NP Classification on Dependent Data. The focus of my own research would be on how to apply the NP paradigm for classification tasks on dependent data, including time series. However, this course project would be on the visualization of the forecasting of time series.
  - Declaration: I hereby declare that this project is done solely within the course.
- PATUPAT Albert John Lalim (20544416)
  - Supervisor: Prof. WONG Raymond
  - Research Topic: Graph Mining, efficient randomized algorithms for Personalized PageRank.
  - Declaration: I hereby declare that this project is done solely within the course.
- WAN Ching Pui (20349359)
  - Supervisor: Prof. CHEN Qifeng
  - Research Topic: Robust Federated Learning. I will explore the time series models in the group project, which I have no experience before.
  - Declaration: I hereby declare that this project is done solely within the course.
- WU Huimin (20349359)
  - Supervisor: Prof. CHENG Tim
  - Research Topic: Noise handling (including label errors and ambiguities). This project will apply a model-agnostic interpretation method to time series prediction models, both of which I have no previous research experience on.
  - Declaration: I hereby declare that this project is done solely within the course.
- ZHANG Shunkang (20580484)
  - Supervisor: Prof. YANG Can and Prof. WANG Yang
  - Research Topic: My research topic is mainly in the field of deep generative learning, style transfer and distributed algorithms. In this project, we will mainly work on time series data.
  - Declaration: I hereby declare that this project is done solely within the course.
- ZHU Zhihan (20707434)
  - Supervisor: Prof. WANG Jiguang
  - Research Topic: Computational Biology, combining genomics data with clinical data.
  - Declaration: I hereby declare that this project is done solely within the course.

1 Introduction

In recent years, deep neural networks have been applied to a large variety of machine learning tasks, producing high-quality results. However, despite their ubiquitous application and outstanding performance, deep learning models are heavily criticized for their black-box nature, which leads to difficulties in applying such techniques in safety-sensitive scenarios such as medical diagnosis [6], or scenarios requiring transparent decision-making such as stock price prediction [10]. To fill in the semantic gap, there is a line of research focusing on the interpretation of deep neural networks [9, 12, 13]. In this implementation project, different time series prediction models will be visualized and explained by a model-agnostic interpretation method. Furthermore, the methodology will utilize datasets from fields with high demands for interpretation, simulating practical applications. Indeed, analysis of the results of this implementation project may help motivate improved model architectures leading to high-quality performances and visual explanations for time series models.

2 Related Work

The black box nature of deep neural networks is a barrier for researchers and practitioners to adopt these models for applications that require more interpretability. In order to explore the mechanisms of a deep neural network and to make it more transparent and explainable, extensive research efforts have been devoted to visual explanation techniques. One of the earliest methods is CAM [16], which is the first to extract information from gradients to capture features importance. It associates the feature maps in the final convolutional layer with particular classes, and afterwards, it uses weighted activations for determining which inputs are important. Extending this approach of CAM, Grad-CAM [13] is applicable to a wide variety of CNN-based models, such as CNNs with fully-connected layers and CNNs with multimodal inputs. Both CAM and Grad-CAM utilize the gradient flowing into the last layer of CNN to assign the importance score. In contrast, DeepLIFT [15] (DEEP Learning Important FeaTures) is an approach based on back propagation. Specifically, it assigns importance scores to the inputs for a given output by collecting the signals from all neurons in the neural network related to the input. Notably, these approaches are mostly applied to models for the image classification task or the general classification task, allowing exploration of visual explanation methods applied to models built for other tasks.

Regarding data, time series data is very common in contemporary life - network traffic, stock price, medical data, etc. In order to improve time series prediction accuracy, several methods are proposed to explore this kind of sequential data. Traditionally, practitioners use statistical methods, such as the Autoregressive Integrated Moving Average Algorithm (ARIMA) [1, 11], to model sequential data. ARIMA introduces a log operator and is combined with an autoregressive component to capture time dependency. The main drawback of this method is that it cannot capture the long-term relationships between the input and output. With the development of Recurrent Neural Networks, researchers propose new architectures Long Short-Term Memory (LSTM) cells [5] and Gated Recurrent Units (GRU) [3]. Both methods introduce a hidden state and a cell state to store the important long-term relationships. Instead of focusing only on univariate time series, MT-GNN [19] deploys a graph neural network to model the latent architecture of a multivariate time series. It proposes a novel mix-hop propagation layer and a dilated inception layer to capture the spatial and temporal dependencies within the multivariate time series. Overall, with this substantial research on time series prediction, it is significant to apply visual explanation methods to models built for the time series forecasting task.

3 Implemented Models and Explanation Methods

3.1 Time Series Forecasting Models

A multivariate time series contains n variables, each of them is associated with its own time series with length T. In this report, we refer the variable as *feature* and the value of a variable at a certain time step as *value*. Time series forecasting is a task to predict values in future time steps given the past values of different features.

We focus on three time series prediction models: Long-and-Short-term Time-series Network (LSTNet) [7], Temporal Pattern Attention LSTM (TPA-LSTM) [14], and Multilevel Wavelet Decomposition Network (mWDN) [17].

3.1.1 LSTNet

LSTNet is one of the earliest works on deep multivariate time series models. It addresses the weakness of neural networks on predicting targets with a large range of scale. LSTNet decomposes the time series into a linear part and a nonlinear part; the linear part is captured by the classical Autoregressive model, while the nonlinear part is parameterized by CNN [8] and GRU [4]. LSTNet uses CNN for its first layer to capture the short-term patterns and the local dependencies between features. Afterwards, the seasonality and the long-term dependencies are captured by two parallel GRUs, one of which having a skip connection.

3.1.2 TPA-LSTM

The LSTNet has two notable shortcomings: 1) the skip length of the recurrent-skip layer has to be manually tuned, and 2) the multivariate time series data are assumed to exhibit a strong periodic pattern.

As an enhancement of LSTNet, TPA-LSTM is capable of tackling time-invariant patterns across multiple time steps, which makes it adaptive to non-periodic and non-linear multivariate time series. A typical attention mechanism for multivariate time series forecasting uses the weighted sum of the column vector of hidden states in the RNN as the context vector. This method identifies the relatively important time steps for all features, and hence, it is not capable of detecting temporal patterns. In contrast, for TPA-LSTM, the attention mechanism uses the weighted sum across the row vectors, which captures the information across multiple time steps for each feature.

To be specific, the first step is to train the LSTM [5] to capture long-term patterns. For the second step, a set of CNN filters were used to extract time-invariant temporal patterns. In the paper proposing this model, the avg-Discrete Fourier Transform was used as the CNN filters to capture the frequency-domain information in multivariate time series. For the third step, the hidden states H and C of the RNN and CNN were aggregated as $H^C = HC^t$. This step filters the result of the RNN by the importance of the frequency-domain from CNN. As the final step, attention was applied to the row vector of H^C.

3.1.3 mWDN

The majority of previous works on time series neural networks, such as RNN [18], LSTM [5], fall into the category of time-domain methods. Time-domain methods take a time series of data points as direct input, wherein correlations between these data points are modelled in order to produce accurate predictions.

Another line of research into time series prediction instead considers the frequency domain. For example, mWDN repeatedly decomposes a time series into several sub-series of high and low frequencies. To be specific, the i-th level of the network applies two convolutional kernels, one with a low pass filter and one with a high pass filter. Afterwards, the output of the low pass filter is fed into the next component, while the output of the high pass filter is fed into the $(i+1)$-th level. With an increasingly higher level of decomposition, the model has an increasingly higher time and frequency resolution.

3.2 SHapley Additive exPlanations

SHapley Additive exPlanations (SHAP) is a unified approach to interpret model predictions, which uses a class of additive feature importance methods. In the original paper proposing SHAP [9], the authors mainly applied the tool to discrete datasets without time dependence. Furthermore, a common variation of SHAP - Deep SHAP - assumes that the input features are independent and that the deep learning model is linear, which typically do not hold when handling time series data. Hence, in this implementation project, different time series forecasting models will combined with SHAP in order to explore, evaluate, and analyze such a strategy.

4 Experiments

4.1 Datasets

This implementation project used four multivariate time series datasets: Solar, Traffic, Energy, and Exchange Rate. The number of features n and the time series length T of each dataset are summarized in Table 1. For each dataset, the time series are split as follows: the first 60% of time steps are assigned

Dataset	Number of features (n)	TS length (T)	Time interval
Solar	137	52560	10 minutes
Traffic	862	17544	1 hour
Electricity	321	26304	1 hour
Exchange rate	8	7588	1 day

Table 1: Summary of datasets: the number of features in the multivariate time series, the total number of time steps, and the time interval between time steps.

to the training set, the next 20% are assigned to validation set, and the last 20% are assigned to the test set.

Solar. This dataset consists of the solar power production records in the year of 2006. It is sampled every 10 minutes from 137 PV plants in Alabama State. The production records of the 137 PV plants was interpreted as a time series with 52,560 time steps. This dataset was used in [7].

Traffic. This dataset consists of a collection of 48 months (2015-2016) hourly data from the California Department of Transportation. The data describes the road occupancy rates (between 0 and 1) measured by different sensors on the San Francisco Bay area freeways. The reading of each of the 862 sensors was interpreted as a time series with 17,544 time steps. This dataset was used in [7].

Energy. This dataset consists of the hourly electricity consumption in kWh recorded from 2012 to 2014. The dataset records the electricity consumption of 321 clients. We will perform multivariate time series forecasting tasks on this dataset. The data for each of the 321 clients was interpreted as a time series with 26,304 time steps. This dataset is available at the UCI Machine Learning repository.

Exchange Rate. This dataset consists of the daily exchange rates between the USA and eight countries, including Australia, British, Canada, Switzerland, China, Japan, New Zealand, and Singapore ranging from 1990 to 2016. The daily exchange rates of each of the 8 countries was interpreted as a time series with 7,588 time steps. This dataset was used in [7].

For each model, the w previous time steps are used to forecast the h-th future time step, wherein w is denoted as window size and h is denoted as horizon. Specifically, $w = 168$ for all datasets, $h = 24$ for Electricity, and $h = 12$ for all other datasets.

4.2 Evaluation Metrics

In order to measure the performance of the time series prediction models, the root relative square error (RSE), the relative mean error (RAE), and the empirical correlation coefficient ($CORR$) of each model were computed:

Root Relative Square Error (RSE):

$$RSE = \frac{\sqrt{\sum_{(i,t)\in\Omega_{\text{Test}}} \left(Y_{it} - \hat{Y}_{it}\right)^2}}{\sqrt{\sum_{(i,t)\in\Omega_{\text{Test}}} \left(Y_{it} - \text{mean}(Y)\right)^2}}$$

Relative Absolute Error (RAE):

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

Empirical Correlation Coefficient (CORR):

$$CORR = \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_t \left(Y_{it} - \text{mean}(Y_i)\right)\left(\hat{Y}_{it} - \text{mean}(\hat{Y}_i)\right)}{\sqrt{\sum_t \left(Y_{it} - \text{mean}(Y_i)\right)^2 \left(\hat{Y}_{it} - \text{mean}(\hat{Y}_i)\right)^2}}$$

where Ω_{Test} denotes the set of indices in the test set, and $Y, \hat{Y} \in \mathbb{R}^{n\times T}$ are the ground truth and predicted time series, respectively.

Dataset	Model	*RSE*	*RAE*	*CORR*
Solar	mWDN	**0.3923**	**0.2439**	**0.9250**
	LSTNet	0.4239	0.2844	0.9094
	TPA-LSTM	0.8408	5.7878	/
Traffic	mWDN	0.8800	0.8685	0.7687
	LSTNet	**0.5071**	**0.3404**	**0.8598**
	TPA-LSTM	1.6133	4.3277	/
Electricity	mWDN	0.1668	0.1267	0.8655
	LSTNet	**0.0995**	**0.0542**	**0.9049**
	TPA-LSTM	0.1487	1.8715	/
Exchange rate	mWDN	0.1060	0.1018	0.8923
	LSTNet	**0.0357**	**0.0296**	**0.9538**
	TPA-LSTM	0.0982	0.3127	/

Table 2: Performance of time series models estimated on four datasets. *CORR for TPA-LSTM is not reported since TPA-LSTM was implemented with a univariate assumption and the comparison between CORR of multivariate and univariate models may not be fair.*

RSE and *RAE* are similar to L_2 loss and L_1 loss, respectively, except that they are normalized. Specifically, their numerators are exactly the L_2 loss and L_1 loss of the model, while their denominator corresponds to the L_2 loss and L_1 loss of a constant estimator, the average of the ground truth values. Hence, *RSE* and *RAE* describe how well the model can outperform the constant estimator, wherein lower *RSE* and *RAE* imply that the model performs better.

CORR is similar to the Pearson correlation coefficient [2], except that *CORR* takes the average Pearson correlation coefficient among the n features of the multivariate time series. *CORR* measures the correlation between the ground truth and predicted values, wherein a higher *CORR* implies that the model performs better.

4.3 Experimental Results of Time Series Models

Table 2 summarizes the performance of LSTNet, TPA-LSTM, and mWDN on each dataset. From the results, mWDN did not perform as well as LSTNet. One possible reason is that in this implementation project, in order to balance the effectiveness and efficiency of the models, only 1-level decomposition for the mWDN was used, which has limited its capability. However, despite this size limitation on the mWDN, the model was too large to be deployed on a GPU, and training 100 epochs on Solar dataset took more than three days to finish.

For TPA-LSTM, it was observed that the trained model was heavily underperforming compared to the original paper. After further investigation, it was found that the implementation in this project did not align with the original paper, wherein the latent structure between the different features was disregarded. Hopefully, an improved implementation may be used for the final presentation.

4.4 Visualization of SHAP

4.4.1 SHAP on LSTNet

Due to limited computing resources, only the first 10 samples in the validation set of the Exchange Rate dataset were used to perform SHAP analysis on the LSTNet model. The absolute SHAP values for each sample and for each feature are summed up and visualized using heatmaps, as shown in Figure 1. Specifically, the x-axis represents the 8 currencies as the input features, the major y-axis represents the 8 currencies as the output predictions, and the minor y-axis represents the 10 samples. A darker color indicates a higher SHAP value, which in turn indicates that a feature is more important for a specific prediction of a specific currency.

From Figure 1, the currencies are strongly pairwise independent and strongly dependent only on its past values, wherein the colors are darker on the main diagonal. To evaluate the validity of these SHAP values, a perturbation analysis was performed: one dataset replaced all values of currency 0 with its mean, and another dataset replaced all values of currency 5 with its mean. Currencies 0 and 5 were selected since they have high and low SHAP values with respect to Currency 4. Afterwards, LSTNet models were trained on these two perturbed datasets. Results show that *CORR* is lower when currency

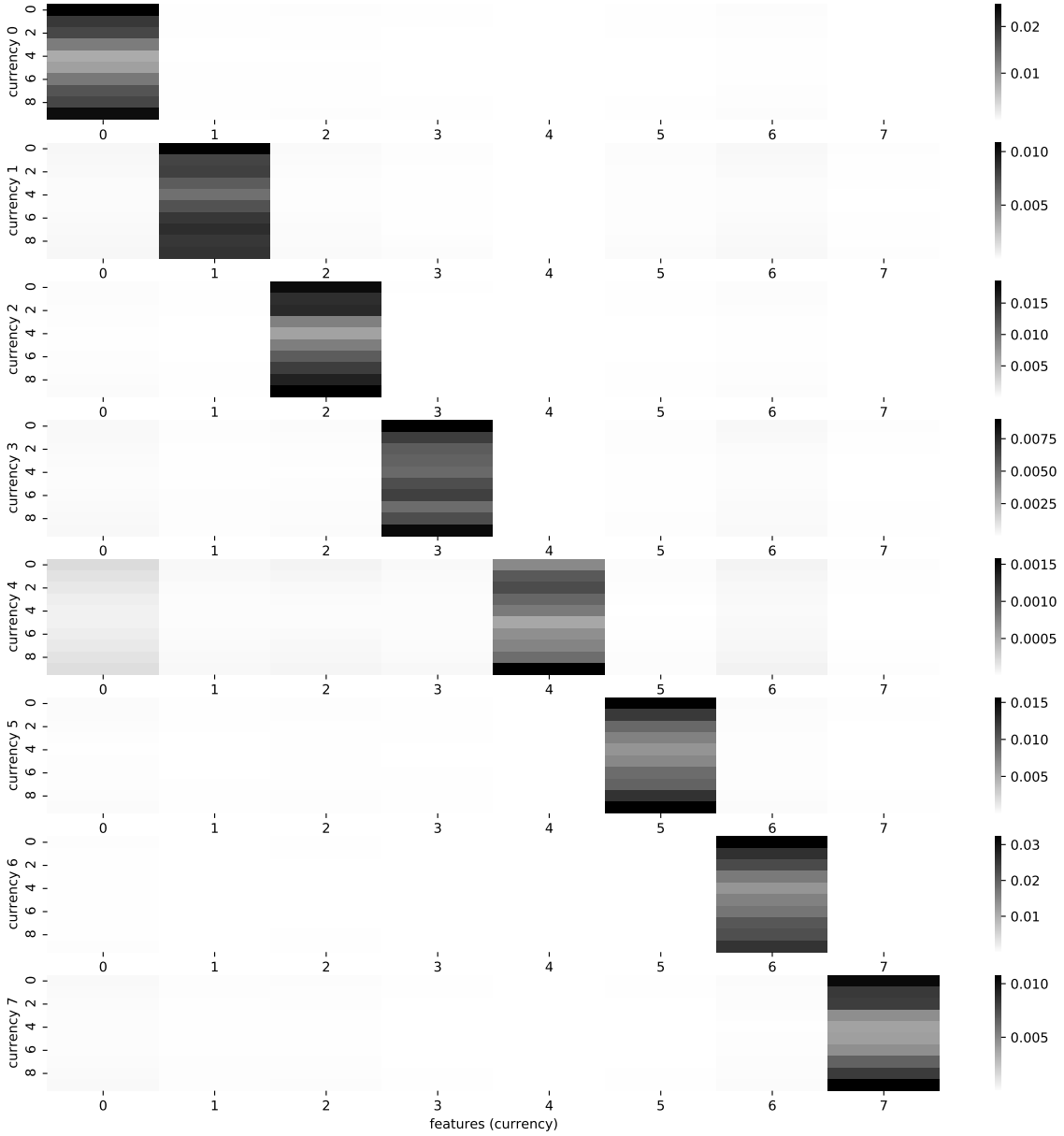

Figure 1: Heatmaps of SHAP values in Exchange rate dataset. The heatmaps describe SHAP values in 10 samples. The columns correspond to the input features. The rows correspond to the output features.

0 is perturbed (0.8905) compared to the original dataset (0.8950), while $CORR$ is higher when currency 5 is perturbed (0.8961). This verifies the SHAP analysis wherein currency 0 is indeed more important than currency 5 in predicting currency 4.

4.4.2 SHAP on mWDN model

Due to limited computing resources, only the first 10 samples in each validation set were used for the SHAP analysis of the mWDN models. Furthermore, this analysis utilized the SHAP values with respect to the entire output, wherein the absolute SHAP values with respect to each output coordinate are summed up.

The first step in this analysis was to identify the most influential time series for each prediction example. This was achieved by summing up the SHAP values of all data points in each time series. As shown in Figure 2 wherein the x-axis represents the input features and the y-axis represents the 10 samples, a small number of time series are truly influential for a given prediction. Hence, this observation allows interpretation to be focused on a smaller set of time series, improving feasibility.

The second step in this analysis was to visualize the most influential time series for a given example, highlighting the most influential data points. As shown in Figure 3 wherein the x-axis represents time and the y-axis represents the top 3 most important features of example 0, a small number of data points in each time series are truly influential. Furthermore, these important data points appear to cluster into intervals. Hence, similarly, this observation allows interpretation to be focused on a smaller set of time intervals, improving feasibility.

The last step in this analysis was to quantify the sensitivity of the SHAP values with a perturbation analysis. This was achieved by selecting the top 10% most influential data points, and replacing each with the mean of the 24 data points before it and after it. Effectively, this method outputs a perturbed version of the input wherein the influential data points are smoothened. Afterwards, the average percent difference between the error of the original input and the error of the perturbed input was computed. Intuitively, a higher average percent difference implies that the most influential points identified are truly influential to the accuracy of the prediction.

Dataset	Exchange Rate	Solar	Electricity	Traffic
Average % Difference	1.8195%	-0.1079%	0.8653%	7.5525%

Table 3: Perturbation analysis on SHAP with mWDN. Intuitively, a higher average percent difference implies that the most influential points identified are truly influential to the accuracy of the prediction.

As shown in Table 3, the result of this analysis is highly dependent on the dataset used. In order to explain this observation, there are three possible reasons: 1) The assumptions of the SHAP values, such as the additive assumption, are not feasible for datasets with highly dependent features, which is typical for time series data. 2) The trained mWDN used in the analysis may not have performed well in the selected 10 examples due to the discussed size limitations. 3) The smoothing method for perturbation may be more effective in destroying features for specific datasets and time series.

In summary, the main advantage of applying SHAP to time series models is that it identifies a sparse number of influential time series and intervals, allowing easier interpretation. On the other hand, the main disadvantage of applying SHAP to time series models is that the assumptions of SHAP may not feasible for all time series data, which are known to be highly dependent and containing long-term patterns.

4.4.3 SHAP on TPA-LSTM model

In terms of our first version implementation, we simply ignore the dependence between different features within each sample. We split each time series along the feature dimension and we regard them as independent time series from different samples, which might be inappropriate. In this way, if we apply SHAP on TPA-LSTM, we cannot find the relationship between each feature and each sample. Due to the limitation of our first implementation, we use Tensorflow to rewrite TPA-LSTM. In this version, we strictly follow the original paper. We use the LSTM to model the latent structure among different features in each sample. What's more, we add the implementation on Temporal Pattern Attention Mechanism. The model is still training and we will show more results on the final representation.

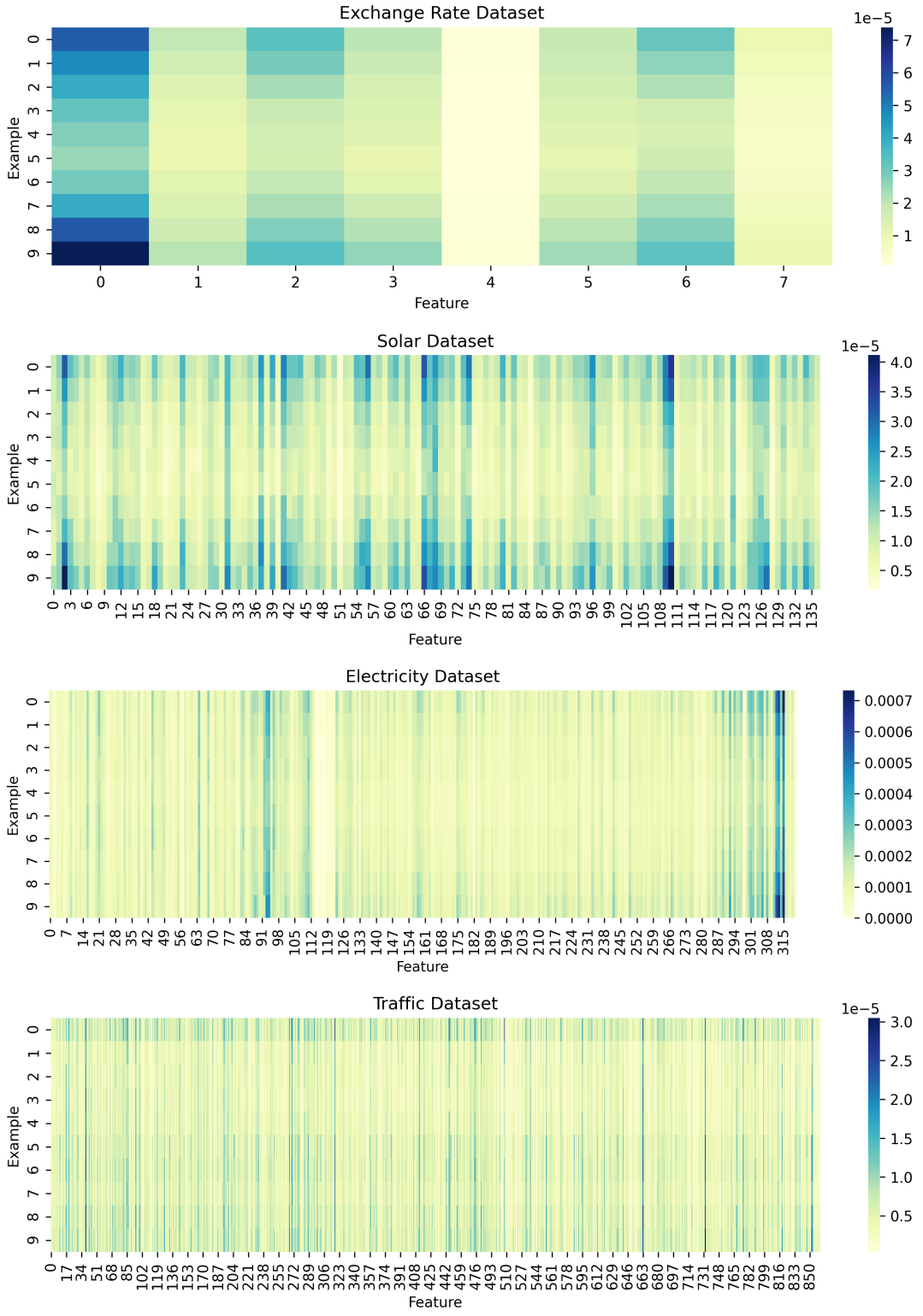

Figure 2: mWDN SHAP values of Features per Example

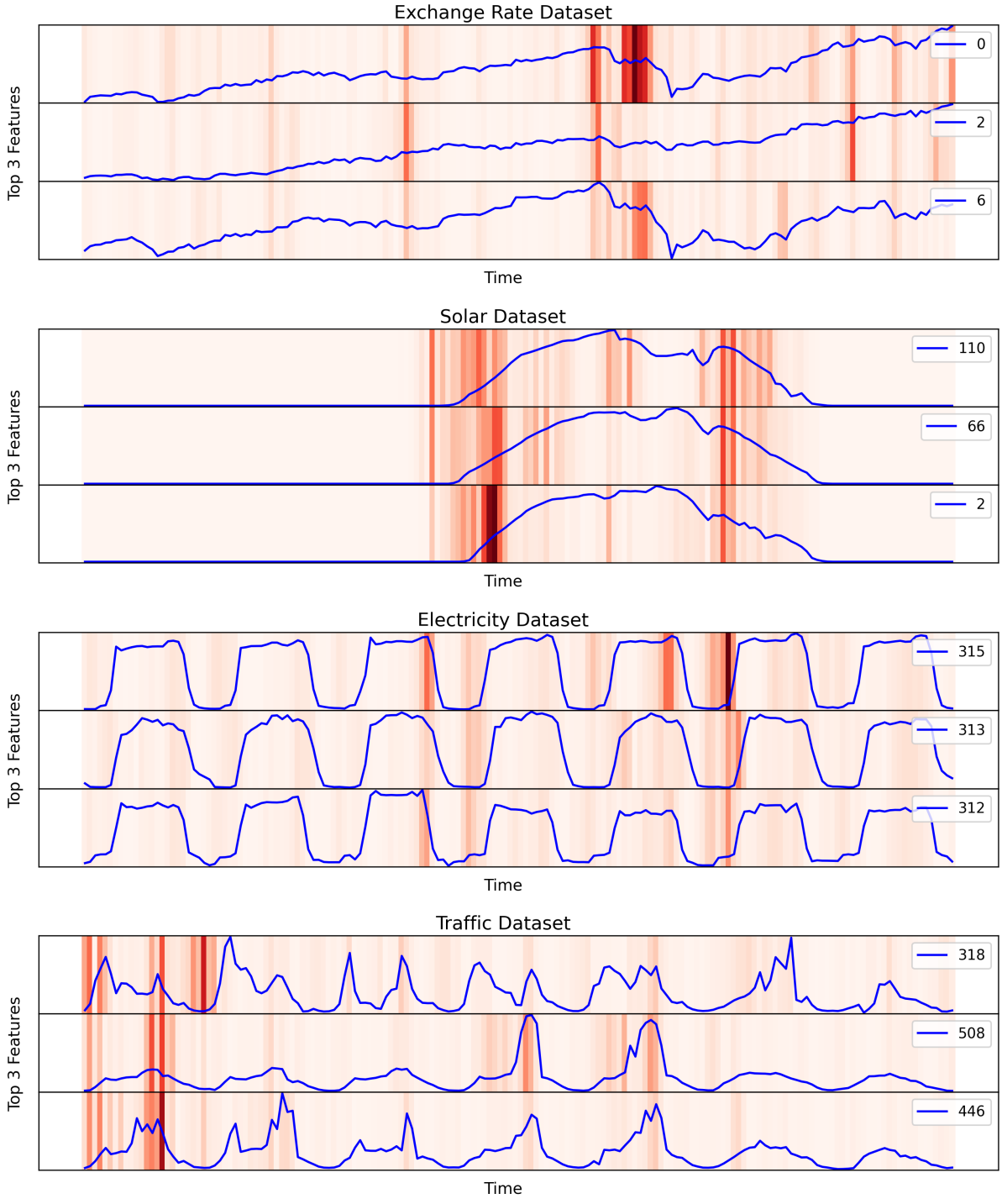

Figure 3: mWDN SHAP values of Top 3 Most Influential Features for Example 0

5 Conclusion

In this implementation project, the experiments featured three deep time series forecasting models: LSTNet, TPA-LSTM, and mWDN. Specifically, these neural network architectures were implemented, trained, and evaluated on four different multivariate time series datasets. In order to interpret these models, their corresponding SHAP values were computed, analyzed, and visualized using heatmaps and shaded line graphs. Using the LSTNet models, perturbation analysis was performed to evaluate SHAP's effectiveness on identifying features important for model training. On the other hand, using mWDN models, perturbation analysis was performed to evaluate SHAP's effectiveness on identifying features influential towards a given prediction of a given model. Overall, the results of the experiments suggest that SHAP may not be suitable for explaining all types of multivariate time series models. Henceforth, future research may focus on interpretation methods feasible for multivariate time series data.

References

[1] Ariyo, A. A., Adewumi, A. O., and Ayo, C. K. Stock price prediction using the arima model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation* (2014), IEEE, pp. 106–112.

[2] Benesty, J., Chen, J., Huang, Y., and Cohen, I. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.

[3] Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[4] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[5] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[6] Kaushik, S., Choudhury, A., Sheron, P. K., Dasgupta, N., Natarajan, S., Pickett, L. A., and Dutt, V. Ai in healthcare: Time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in Big Data 3* (2020), 4.

[7] Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (New York, NY, USA, 2018), SIGIR '18, Association for Computing Machinery, p. 95–104.

[8] Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks 8*, 1 (1997), 98–113.

[9] Lundberg, S. M., and Lee, S.-I. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2017), NIPS'17, Curran Associates Inc., p. 4768–4777.

[10] Mokhtari, K. E., Higdon, B. P., and Başar, A. Interpreting financial time series with shap values. In *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering* (USA, 2019), CASCON '19, IBM Corp., p. 166–172.

[11] Nochai, R., and Nochai, T. Arima model for forecasting oil palm price. In *Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and applications* (2006), pp. 13–15.

[12] Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), KDD '16, Association for Computing Machinery, p. 1135–1144.

[13] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 618–626.

[14] Shih, S.-Y., Sun, F.-K., and Lee, H.-y. Temporal pattern attention for multivariate time series forecasting. *Machine Learning 108* (06 2019).

[15] Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685* (2017).

[16] Sundararajan, M., Taly, A., and Yan, Q. Gradients of counterfactuals. *arXiv preprint arXiv:1611.02639* (2016).

[17] Wang, J., Wang, Z., Li, J., and Wu, J. Multilevel wavelet decomposition network for interpretable time series analysis. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (New York, NY, USA, 2018), KDD '18, Association for Computing Machinery, p. 2437–2446.

[18] Williams, R. J., and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput. 1*, 2 (1989), 270–280.

[19] Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., and Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. *arXiv preprint arXiv:2005.11650* (2020).

Member Contributions

LIU Sin Tai

First, I did literature review for time series and explainable AI related project. Second, I am responsible for the implementation and training of the TPA-LSTM model. I had to align the code that we have for TPA-LSTM with the rest of the models that we are using for the fair comparison. It was quite a bit of work, cause the structure and methodology of our implementation for TPA-LSTM was inherently different from the rest of the group, which requires requires a lots of studies and understanding to do the modification. Thirdly, I will apply SHAP on TPA-LSTM with ZHANG Shunkan.

PATUPAT Albert John Lalim

My contributions towards this project include the following: 1) I contributed heavily in identifying the project topic, and in planning which models and interpretation methods to implement. 2) I was mainly responsible for SHAP on the mWDN models, which included the time-consuming SHAP calculation on all four mWDN models, the design and implementation of the visualizations and experiments on the SHAP values, and the accompanying analysis of these SHAP values. 3) I contributed heavily in writing, proofreading, and formatting the proposal and the final report.

WAN Ching Pui

I prepared the datasets, implemented and trained the LSTNet, designed visualization for SHAP values, and took up the project management role. I collected the five datasets (four in this report and one that we did not have time to evaluate) and made an API for accessing the data. I implemented the LSTNet in PyTorch. My implementation was based on the original code: I adapted the data processing and evaluation metrics, and upgraded the framework to the latest PyTorch version. I trained and tuned the LSTNet on the four datasets. Besides, I wrote the visualization codes for plotting time series with the SHAP values as the background intensity. I also did the project management: setting up git repositories and organizing zoom meetings.

Wu Huimin

I am responsible for implementation of mWDN and get well-trained model ready for further visualization and explanation. More specifically, I adapted mWDN from original repo to four dataset-of-interest by: 1) figuring out correct version of environment including cuda, PyTorch, etc; 2)splitting out mWDN and its dependencies from existing implementation and 3) fixing all errors caused by the adaptation. Also I trained this model on four datasets, evaluated it performance and handed the sufficiently-trained model to Albert for visualization. Finally, I contributed to the writing of description of mWDN and part of organization of the paper structure.

ZHANG Shunkang

In this project, I first involve the discussion of the topic. I am not very familiar with time series data and I read a lot of related articles. Based on these literature reviews, I finish the related work section in final report. I also help train TPA-LSTM and figure out the difference between the implementation and original paper. I retrain the TPA-LSTM model by using tensorflow based on the reference code. In this version, we capture the dependent features in each sample. With this well trained model, I apply SHAP on TPA-LSTM and do the visualization with LIU Sin Tai. I finished the corresponding part in the final report.

ZHU Zhihan

I calculated the SHAP values in the exchange rate dataset for the LSTNet model using a python package named shap and transformed the SHAP values into a different form by summing up the absolute values, which is easy to visualize and interpret. After finishing transformation, I drew the heatmap of the SHAP values and designed the plan to evaluate the explainable model. To figure out if the feature which has a high SHAP value really matters in the LSTNet model, I generated two modified exchange rate datasets according the SHAP values, and with the help from WAN Ching Pui, I finished the evaluation of the SHAP model on LSTNet model and wrote the corresponding part in the final report.