Multivariate Time Series Forecasting using Transformers

Parth Patel, Muralidhar Koripalli and Kratika Shetty
Worcester Polytechnic Institute, USA
pdpateli@wpi.edu,mkoripalli@wpi.edu,kshetty2@wpi.edu

ABSTRACT

A Multivariate Time Series is made up of complicated combination of inputs. It can have more than one time-dependent variable and each variable depends not only on its past values but also has some dependency on other variables. Along with this, Time series data can exhibit short term dependencies or long term dependencies based on the level statistical dependence between two points in the time series. Time series forecasting model should have the ability to efficiently capture exact long-range dependencies and temporal dynamics between its inputs.

Transformer has been proven in recent research to have the ability to improve prediction capacity in Time Series Forecasting. But Transformers cannot be directly applied to Time Series due to its quadratic time complexity, high memory utilization, and inherent limitation due to its encoder-decoder architecture. In this project, we implemented Temporal Fusion Transformers which is efficiently able to capture temporal dynamics by selecting only relevant features and suppressing unnecessary components. TFT model had significant performance on time series data with short term dependencies. Informer proposed by Haoyi Zhou et al. in the paper Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting also was implemented to improve the results on long term dependencies. Informer was able to improve the performance and increase the performance of our model.

1 INTRODUCTION

Time Series forecasting is a concept of forecasting the future trends based on the past data, and has been applied in various domains one of them is future sales prediction. Forecasting problems consists of combination of inputs – time-invariant variables which are known as Static Covariates. In the case of the sales prediction, Store Location, Product Information are few of the variables that can be considered as static covariates. Sales are also affected by known future inputs such as holidays and information about shop closures. In addition to these inputs, we can be given time series data from the past (for example, historical consumer foot traffic) with no prior knowledge of how this data affects sales. Time series forecasting is particularly difficult due to different inputs available and the lack of knowledge about their input relationships.

Deep neural networks are increasingly being employed in multi-horizon forecasting, and they have shown to outperform classical time series models. While many models based on Recurrent Neural Networks were initially designed, recent advancements have also incorporated attention-based models to increase the selection of important time steps in the past.

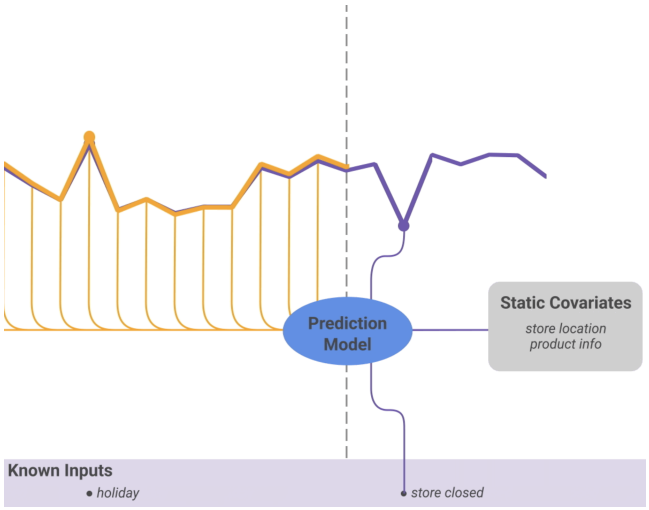

Figure 1: Multivariate forecasting with static covariates and various time-dependent inputs.

Attention-based models can identify relevant time steps. However, these models cannot provide insights on how important inputs features are at different time step. Temporal Fusion Transformer model is able to overcome this limitation in transformer by making the forecasting interpretable. For local processing, TFT uses recurrent layers, whereas for long-term dependency, it uses interpretable self-attention layers. But in the case of long term dependencies, we observed that Informer outperformed TFT.

Encoder in Informer is designed to extract long-range dependencies of time series data. It is also efficiently able to handle extreme long input sequences by using ProbSparse self-attention mechanism which is followed by distilling operation. It enables the encoder to highlight dominating attention by halving cascading layer input. The long time-series sequences is predicted at the decoder in one forward operation instead of a step-by-step way. This helps improve the inference speed of long-sequence predictions.

1.1 Research Contributions

In this project, we compared the performance of LSTM, TFT and Informer. Informer architecture was implemented using the code available for the Informer paper. We analysed the dataset and observed that there are two types of dependencies present in the time series. We came up with an approach that could handle both the types of dependencies. We concluded that a combination of TFT and Informer would improve the performance of our model.

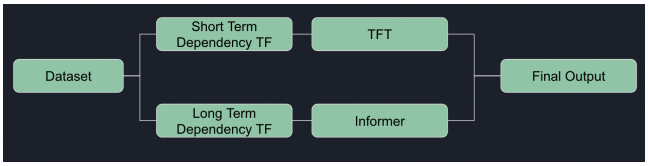

Figure 2: Solution Proposed

2 BACKGROUND

Time-series forecasting methods could be roughly grouped into two categories - Classical Statistical Time Series Models and Deep Learning Model using RNN networks (Li et al. 2018; Yu et al. 2017). But RNN based networks do not have good performance on long sequence time series.

Attention based models were developed to overcome the problem of long sequences and capture the importance of certain periods of time that occur periodically and can impact the forecast. There have been some previous studies on how to improve the efficiency of self-attention - The Sparse Transformer (Child et al. 2019), LogSparse Transformer (Li et al. 2019), and Longformer (Beltagy, Peters, and Cohan 2020). But these transformers perform well only on extremely long sequence time series.

3 PROPOSED METHOD

We compared and implemented 3 models LSTM, Informer and TFT. Our final model is a combination of the 2 attention models mentioned above.

3.1 Data & Preprocessing

The data we considered for this work is a Kaggle competition data set. The data include sales of products of 33 classes at 54 different stores, store details and other factors: holidays, transactions, oil prices, which have an impact on the store sales. After Analysing the data we found that, there are 2 different sales patterns in the past. Pattern with Long term dependency and pattern with short term dependency. The Sales of the product class 'SCHOOL AND OFFICE SUPPLIES' and 'FROZEN FOODS' are the only data that requires looking far back to make a prediction.

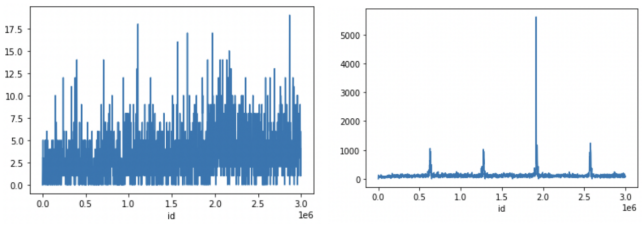

Figure 3: Plot on the left of Sales vs Time Index for Store Number 1 and Product Family = Automotive exhibits short term dependency. Whereas, plot on the right of Sales Vs Time Index of Store Number 1 and product family = Frozen Foods exhibits long term dependency.

3.2 Design

LSTM: This model involves processing of the sale history of different products at different stores as individual time series and predicting them as if they were uni-variate time series. This implementation only considers the sales data does not give any importance to the oil prices and holidays.

Temporal Fusion Transformers: TFT was implemented using Pytorch Forecasting framework. The major features of TFT are -

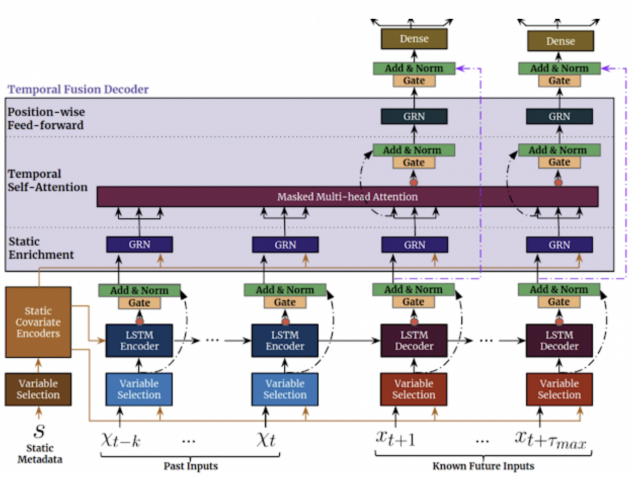

Figure 4: Inputs static metadata, time-varying past inputs and time-varying known future inputs are fed into TFT. Salient features are selected through variable selection. Gated information is added as a residual input which is followed by normalization. GRN blocks enable efficient information flow. LSTMs are used for local processing and information from any time step is integrated using multi-head attention.

- Gating mechanisms - It learns from the data to skip unnecessary components providing adaptive depth and network complexity to suit a broad range of datasets.
- Variable selection networks - At each time step, relevant feature is selected. Most of the conventional Deep Neural Network Models overfit to features which are not important in the model. Whereas attention-based variable selection in TFT improves generalization in the model by encouraging the model to focus most of its learning capacity on the most important features.
- Static Covariate Encoders - Static features like store location has significant impact on the sales. TFT makes use of the static metadata and integrates these features to control how temporal relationships are modeled
- Temporal processing to learn temporal relationships between its input is used. A sequence-to-sequence layer is used for local processing. Long-term dependencies are extracted using a multi-head attention block. Using the attention block, it can focus on the time step which would be relevant in the prediction.

- Prediction intervals show quantile forecasts. It displays range of target values instead of single prediction. This helps users understand the distribution of the output, not just the point forecasts.

Informer: The code for the paper informer was utilised to understand the architecture. The code was modified as per our dataset. Encoder of the informer is the distinguishing feature in the Informer architecture. The structure consists of several Attention blocks, Conv1d, and MaxPooling layers to encode the input data. Replicas of the main stack divide the inputs into half to increase the reliability of the distilling operation. In addition, the number of self-attention distilling layers is continually lowered one by one. At the end of the encoder, concatenated Feature Map is fed to the decoder. Decoder structure includes a stack of two identical multi-head attention layers and output elements are predicted in a generative style.

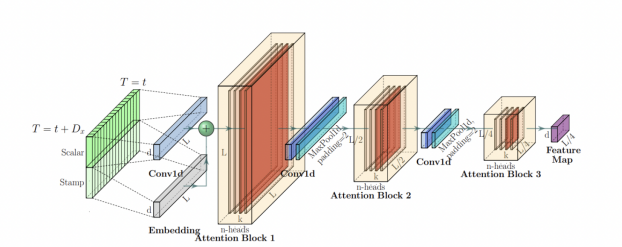

Figure 5: Informer Encoder

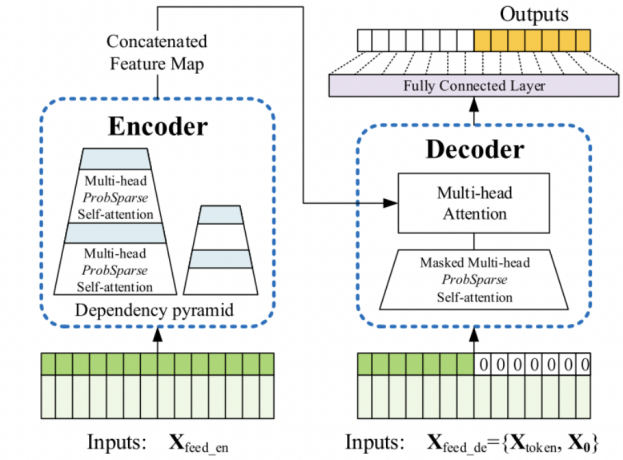

Figure 6: Informer Architecture Overview - The encoder receives the long sequence input. The self attention is replaced with the proposed ProbSparse attention. Dominating attention is extracted in the distilling process which reducing the network size sharply. Robustness is increased by stacking the layer replicas. The decoder receives concatenated feature maps, zero padding is done in the target element. The weighted attention composition of the feature map is measured and output elements are predicted in a generative style.

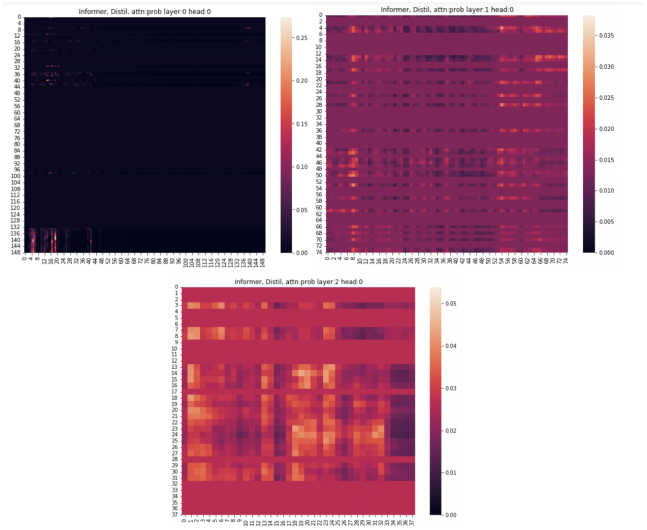

Figure 7: Visualising the Attention Layer - Output of each attention block is displayed above. Attention layer 2 picks only the important information and passes it on to the decoder. We can observe that output of the attention layer keeps decreasing by half.

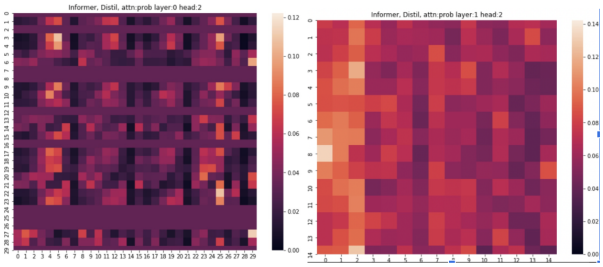

Figure 8: Visualising the Attention Layer - In case of short term dependency, the model tends to loose information since the attention layer focuses only the most important information.

4 EXPERIMENT

We considered the last 15 days of the training data for testing. Each of the models used the last 15 days for testing and the models were evaluated based on its performance on the test data.

4.1 Experimental setup

We used Google Colab Pro platform to run our implementation with a run-time set up with GPU and High RAM.

4.2 Evaluation Metrics:

RMSE, MSE, MAE and RMSLE are calculated for the models.

4.3 Results

Results of LSTM, TFT and combination of TFT and Informer is displayed below. LSTM performed better for data with short term dependencies compared to the data with long term dependencies.

mae	mse	rmse	rmsle
94.947736	110786.211833	332.845628	0.615739

Figure 9: : LSTM Results

	instance_model	mae	mse	rmse	rmsle
0	TFT	87.466494	93076.022517	305.083632	0.591563
1	TFT + Informer	83.945107	89838.131219	299.730097	0.603606

Figure 10: Results for TFT and TFT + Informer

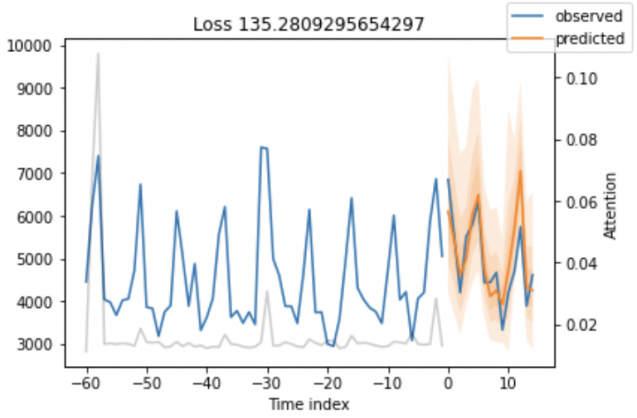

Figure 11: Plot of Sales vs Time Index observed for the combined model. We can see that the observed and predicted values are very close to each other. The plot in grey shows the attention at each and every time step.

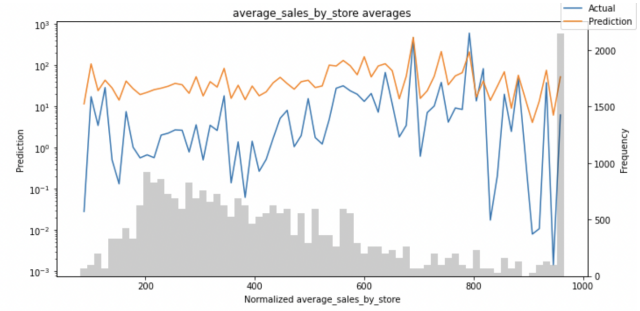

Figure 12: Plot of Normalised Average Sales Per Store

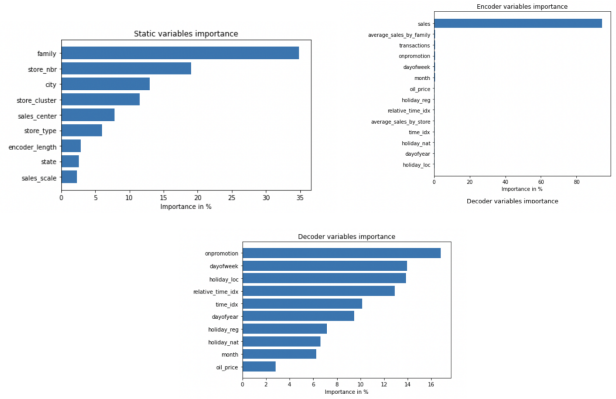

Figure 13: Variables importance given in the past input is represented by Encoder Variables importance. Decoder variables importance shows the importance given to the known future inputs. These are used while making the predictions in the future. Hence TFT is not a black-box model, with attention weights and importance we can find which features are important.

MSE, MAE and RMSE of the combined model is lesser than of TFT model. RMSLE is higher for the combined model compared to TFT model. This disparity might be attributable to the fact that RMSLE penalizes underestimating of the Actual variable more severely than overestimation. RMSLE measure takes into account the relative error between the predicted and actual value, and the magnitude of the error isn't relevant. The RMSE value, on the other hand, increases in magnitude as the scale of error grows. In Business forecasting scenarios, RMSLE would have been a better metric but since in our project we want the model to predict values closer to the true values, we would be considering MAE, RMSE and MSE as the metric. Hence the results suggest that combined model of TFT and Informer gave better results compared of that of just TFT.

5 DISCUSSION

In this project we have implemented Informer Model on sales data of Stores and Product family exhibiting long term dependency. However, we were unable to train the model Informer on the whole dataset due to its high computational requirement. We were unable to find the performance of Informer and compare it with other models.

6 CONCLUSION

Our objective in the project was to analyse how transformer architecture could be leveraged in time series data. TFT architecture proposed by Bryan Lim et al. was implemented and we investigated the temporal relationships learnt. Attention mechanism was used to find relevant features and time steps in the time series data. TFT works especially when time series has short term series dependencies. Informer was able to use its attention block where it is able to capture long term dependencies.

REFERENCES

[1] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[2] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509* (2019).

[3] Jake Grigsby, Zhe Wang, and Yanjun Qi. 2021. Long-Range Transformers for Dynamic Spatiotemporal Forecasting. *arXiv preprint arXiv:2109.12218* (2021).

[4] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems* 32 (2019).

[5] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[6] Bryan Lim, Sercan O Arik, Nicolas Loeff, and Tomas Pfister. 2019. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv preprint arXiv:1912.09363* (2019).

[7] Rose Yu, Stephan Zheng, Anima Anandkumar, and Yisong Yue. 2017. Long-term forecasting using higher order tensor RNNs. *arXiv preprint arXiv:1711.00073* (2017).

[8] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI.*