Beyond Words: Adapting NLP Methodologies for Time Series Forecasting Challenges

Raman LNU
lnu.raman@northeastern.edu
And
Edward Burke
burke.e@northeastern.edu
Team 6

Abstract

Although Natural Language Processing (NLP) methods are highly effective in handling textual data, their potential for handling other kinds of data, like time series, is still largely unexplored. The effectiveness of several NLP models for time series forecasting is examined in this work. We assess five models, ranging in architecture from transformers to convolutional networks: Autoformer, DLinear, WaveNet, DeepAR, and SimpleFeedForward. WaveNet exhibits superior forecasting performance with a Minimal Absolute Scaled Error (MASE) of 1.163, highlighting the usefulness of NLP techniques in time series analysis. Our results demonstrate the versatility and the efficacy of NLP techniques in identifying complex patterns in time series data. A synergistic combination of attention techniques is proposed to improve model flexibility and accuracy through additional research on sparse-dense attention mechanisms. This novel approach has the potential to open up new directions for advanced and flexible time series forecasting techniques.

1 Introduction

The goal of those who pioneered natural language processing (NLP) was to comprehend, process, and produce written text. Transformer architectures and deep learning models have been essential to NLP's advancement. Data scientists have discovered that as NLP technology has advanced, its underlying mathematics have become sufficiently strong to allow NLP models to be used in a wide range of applications, including the creation of music and image classification. Our project combines these concepts with a common problem that arises in many different fields: projecting future outcomes through modelling that is based on historical performance. In this study, we investigated whether NLP systems could be applied to solve such issues and produce practical forecasts using time-series data.

At its core, a time series is a sequence of data points collected or recorded at observable intervals – broadly speaking, a set of data that can be represented on a temporal axis. These can range from stock market fluctuations and economic indicators to weather patterns and energy consumption data. What makes such datasets useful is their chronological order. Unlike random data points, a time series provides historical context, allowing us to analyze and forecast trends, seasonal variations, and cycles.

We will seek to demonstrate that NLP systems can be used for time series and to compare the performance of several different models on a particular dataset. We will begin by explaining how we assembled a proper dataset for our comparative analyses. We will then describe the methods used to select and assess the five models we tested. Finally, we will present and discuss our findings and offer conclusions and thoughts on future avenues of study.

2 Background

Considerable progress and paradigm changes have occurred in the development of time series forecasting techniques, most notably with the advent of transformer models. In the past, statistical models like ARIMA (AutoRegressive Integrated Moving Average) dominated time series analysis. Because of their resilience, these models were essential for comprehending and forecasting time series data by seeing patterns like seasonality and trends.

But it became more and clearer how inadequate these conventional techniques were for dealing with large-scale, complicated, non-stationary, and non-linear time series data. The introduction of transformer models, which were initially created for natural language processing (NLP), addressed this difficulty. Transformers, introduced by Vaswani et al. in their landmark paper "Attention Is All You Need," transformed natural language processing (NLP) by utilising self-attention mech-

anisms to process sequential input efficiently and capture context in a manner that was not possible for earlier models.

Transformers were modified for time series forecasting as a result of this breakthrough in NLP. Transformers are especially well-suited for this field since they are excellent at capturing the dynamism and temporal dependencies found in sequential data, which is similar to what time series analysis demands. As a result of this shift, numerous cutting-edge transformer-based models that each focus on a different facet of time series forecasting have been created.

For example, the Autoformer model adds a decomposition mechanism for trend and seasonal components, while the Informer model's ProbSparse self-attention mechanism is intended for processing lengthy sequences. The FedFormer supports federated learning environments, whereas the Pyraformer uses a pyramidal structure for multi-scale representation. Furthermore, logarithmic sparse attention is used by the LogTrans model for effective long sequence predicting.

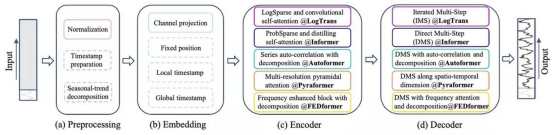

Figure 1: Overview of Transformer Based Models

Furthermore, the most recent developments in time series forecasting have been marked by the incorporation of domain-specific expertise, hybrid models that blend deep learning with statistical techniques, and a heightened emphasis on interpretability and explainability.

3 Methodology And Implementation

Five different NLP models were carefully chosen for our study based on their individual technological and methodological advantages, and we will examine each model's efficacy in time series forecasting. We have chosen the following models: Autoformer, a transformer-based model specifically tailored for time series; DLinear, which focuses on linear forecasting using sophisticated NLP techniques; DeepAR, a recurrent neural network model skilled at capturing temporal dynamics; SimpleFeedForward, a convolutional network originally intended for audio processing, and WaveNet, a convolutional network repurposed for time series analysis.

To comprehend their individual and combined effects on forecasting accuracy, these models—which range in approach from transformers to neural networks—were thoroughly documented and examined.

3.1 Autoformer

Autoformer's unique transformer-based architecture and effective implementation are key factors in its effectiveness in time series forecasting. Among its most important architectural elements are:

- **Encoder-Decoder Architecture**: The encoder-decoder architecture used in transformer models is what Autoformer adheres to. The encoder extracts features and records long-range dependencies from the input sequence, which consists of historical observations. These features are then used by the decoder to produce predictions for upcoming time steps.

- **Self-attention Mechanism**: The self-attention mechanism is the central component of Autoformer. The model is able to discover intricate relationships and dependencies within the data thanks to this mechanism, which enables each element in the input sequence to pay attention to every other element. This is especially helpful in capturing seasonalities and long-range patterns.

- **Hierarchical Transformer Encoder**: A hierarchical transformer encoder with varying resolutions is used by Autoformer. As a result, the model can identify global trends spanning all time series as well as local trends within specific time series. The model can learn intricate patterns and produce precise forecasts more easily thanks to this hierarchical approach.

- **Feature Engineering**:Together with the time series data, Autoformer allows for the integration of external features. These characteristics can give the model more context and data to work with, which will improve its forecasting abilities even more.

In terms of implementation, Autoformer uses the following processes:

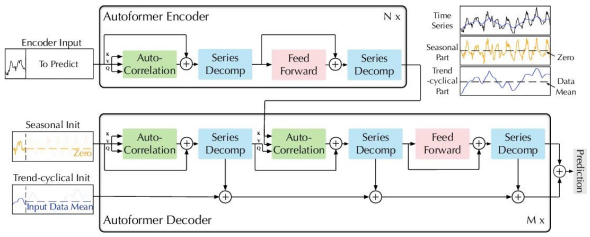

Figure 2: Autoformer Architecture

- **Data Preparation**:As with other models, preprocessing the data is essential. Scaling, normalisation, and missing value imputation are some of the tasks involved in making sure the data is appropriate for Autoformer training.
- **Encoder Processing**: The encoder receives the preprocessed data and feeds it through a series of layers with self-attention mechanisms. Every layer gains the ability to represent the connections between various sequence elements and extract features.
- **Decoder Processing**:The decoder receives its input from the encoder outputs. After that, the decoder makes predictions for upcoming time steps using these characteristics along with its own self-attention mechanisms.
- **Probabilistic Forecasting**:In addition to point forecasts, Autoformer has the option to offer probabilistic forecasts. This enables the model to express the degree of uncertainty surrounding its forecasts, providing a more thorough comprehension of possible future results.
- **Prediction and Evaluation**:Autoformer can be used to predict future time steps once it has been trained. Depending on the needs of the user, the forecasts can be probabilistic or point forecasts. Furthermore, GluonTS offers instruments for assessing the Autoformer model's performance on the held-out dataset.

All things considered, Autoformer's transformer-based design, self-attention mechanism, and hierarchical encoder, along with GluonTS's capabilities, combine to make it a potent and adaptable time series forecasting tool. It is appropriate for a variety of forecasting tasks and datasets due to its capacity to capture long-range dependencies, incorporate external features, and produce probabilistic forecasts. Furthermore, GluonTS streamlines the entire workflow, resulting in an effective and user-friendly process.

3.2 DLinear

Combined with GluonTS's capabilities, DLinear—a new method created especially for time series forecasting—offers a potent and effective solution. Here's how the architecture functions:

- **Context Embedding**: In order to capture the overall trend and seasonality within the time series, this module learns a latent representation of the entire context window.
- **Linear Seasonal and Trend Layers**: These layers project the context embedding to a different space where linear transformations explicitly capture the seasonal patterns and linear trend.
- **Probabilistic Head**: This layer allows probabilistic forecasting by mapping the latent representations from linear layers and context embedding to the parameters of a selected distribution.

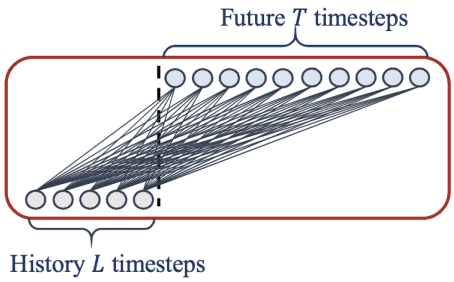

Figure 3: Illustration of Basic Linear Model

As for implementation, it uses the following principles:

- **Data Preparation**: As with other models, loading, pre-processing, and formatting the data into a format appropriate for DLinear training are all part of the data preparation process.
- **Model Construction**: The dimensions of the latent representations and the size of the context window can be specified by users along with the desired DLinear architecture.
- **Training and Optimization**:GluonTS allows for automatic hyperparameter tuning and

seamless integration with a variety of optimisation algorithms, facilitating effective training and ideal configuration for precise forecasting.

- **Prediction**: DLinear can produce point forecasts for upcoming time steps after it has been trained. It can also produce probabilistic forecasts, which put an estimate on how uncertain its predictions are.
- **Evaluation**: GluonTS provides a range of metrics and visualisation tools for assessing the DLinear model's performance on the held-out dataset.

When used in conjunction with GluonTS, DLinear becomes an invaluable tool for forecasting time series. Its unique architecture, which was created with the intention of capturing trends and seasonalities, together with its effective implementation and capacity for probabilistic forecasting, make it an excellent option for a variety of forecasting tasks. GluonTS streamlines the procedure even more, enabling practitioners and researchers to utilise DLinear with ease and accessibility.

3.3 DeepAR

DeepAR's creative strategy and effective execution are the main reasons for its success in time series forecasting. Its architecture functions in this way:

- **Recurrent Neural Network (RNN)**: DeepAR's recurrent neural network layer, which successfully extracts the temporal dependencies from the time series data, is its central component. Accurate predictions are made possible by the RNN's ability to learn how past values affect future values through sequential data processing.
- **Probabilistic Forecasting**: In addition to producing point forecasts, DeepAR also calculates the probability distribution that goes along with those forecasts. This method gives users a more thorough grasp of possible future outcomes while taking into account the inherent uncertainty in forecasting.
- **Global Model**: DeepAR makes use of a global model as opposed to traditional models that create unique models for every time series. By utilising information from all available time series data, this model improves its ability to generalise to new data and learn more intricate patterns.
- **Feature Engineering**: To improve its forecasting abilities, DeepAR uses feature engineering techniques. This may entail combining data from outside sources, such as weather reports or holidays, which can have a big influence on particular time series.

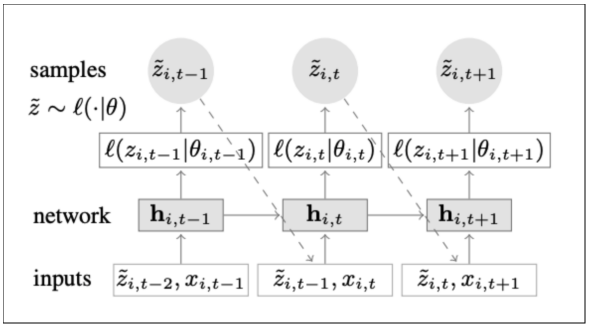

Figure 4: Mathematical Architecture of DeepAR

The implementaiton functions as follows:

- **Data Preparation**: Data is preprocessed using techniques like scaling, normalisation, and missing value imputation before being fed into DeepAR. This guarantees that the data is appropriate for the model's processing and education.
- **RNN Processing**: The RNN layer receives the preprocessed data and processes it in a sequential manner. The RNN picks up the ability to represent the temporal relationships and extract features from the data.
- **Probabilistic Forecasting**: To estimate the probability distribution connected to the predictions, DeepAR makes use of extra layers. This enables the model to express the degree of uncertainty surrounding its projections, offering a more sophisticated perspective on potential futures.
- **Global Learning**: The global model parameters are updated by combining the knowledge gathered from all time series data. The model can learn intricate patterns and generalise well to previously unseen data thanks to this process.
- **Prediction and Evaluation**: DeepAR can be used to predict future time steps once it has

been trained. Probability distributions are included with the forecasts so that users can evaluate the forecasts' level of confidence.

When combined with GluonTS's features and effective implementation, DeepAR becomes a potent tool for time series forecasting. Its global model approach, probabilistic forecasting capabilities, and RNN architecture allow for reliable and accurate predictions. Moreover, GluonTS streamlines the entire process, facilitating easy access to DeepAR for both practitioners and researchers.

3.4 SimpleFeedForward

When paired with GluonTS, the feedforward neural network (FNN) provides useful capabilities for time series forecasting, even though it is less complex than other deep learning models. In particular, SimpleFeedForward was utilised to examine this kind of model. Among its functional and architectural characteristics are:

- **Input Layer**: The time series provides the context window, or historical observations, to this layer.
- **Hidden Layers**: These tiers identify intricate connections within the data and extract relevant features. The model's ability to learn intricate patterns depends on the number of hidden layers and neurons in each layer.
- **Output Layer**: The estimated values for upcoming time steps are generated by this layer.

Implementation is handled as such:

- **Data Preparation**: As with other models, preparation of the data is crucial. To do this, the time series data must be loaded, preprocessed, and formatted so that it can be used for FNN training. Tools from GluonTS make this process easier.
- **Model Construction**: By defining the number of hidden layers, the number of neurons per layer, and the activation functions to be used, users can easily specify the desired FNN architecture. The pre-built activation functions that GluonTS provides, such as sigmoid and ReLU, enable the model's behaviour to be customised.
- **Training and Optimization**: Automatic hyperparameter tuning is possible with GluonTS, which also integrates smoothly with a variety of optimisation algorithms. This makes it possible for users to effectively train the FNN and identify the ideal setup for precise forecasting.
- **Prediction**: The model can produce point forecasts for upcoming time steps after it has been trained.
- **Evaluation**:A range of metrics and visualisation tools are available in GluonTS to assess the model's performance on the held-out dataset.

When paired with GluonTS, FNNs provide a straightforward but effective method for time series forecasting. Their adaptability, interpretability, and computational efficiency make them appropriate for a wide range of datasets and forecasting tasks. GluonTS streamlines the process even more, enabling researchers and practitioners looking for reliable and accurate forecasting solutions to easily access FNNs.

3.5 WaveNet

WaveNet's distinct architecture and operating principles are the reasons behind its efficacy in time series forecasting:

- **Dilated Causal Convolutions**: The dilated causal convolution is the fundamental unit of WaveNet. By using dilated kernels, which have filters placed farther apart, the receptive field is exponentially expanded without requiring an increase in the number of parameters. This preserves resolution while enabling the model to represent long-range dependencies within the sequence.

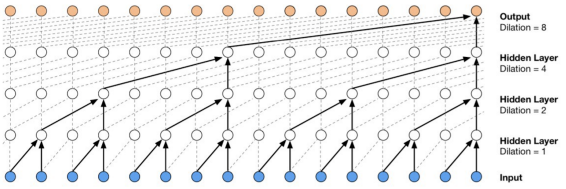

Figure 5: Visualisation of Dilated Causal Convolution

- **Residual Stacking**: WaveNet uses residual connections to add the original input to the output while avoiding layers. This enhances

information flow throughout the network and solves the vanishing gradient issue, especially for lengthy sequences.

- **Gated Activation Units (GLU)**: GLU, a two-branch gating mechanism with a linear branch and a sigmoid gate, is used by WaveNet. By learning to modulate the linear branch's output, the gate enables the model to suppress noise and concentrate on pertinent data.
- **Multi-scale Dilations**: In order to capture patterns at various time scales, WaveNet uses multiple layers with variable dilation rates. As a result, the model can discover both internal and external dependencies in the data.

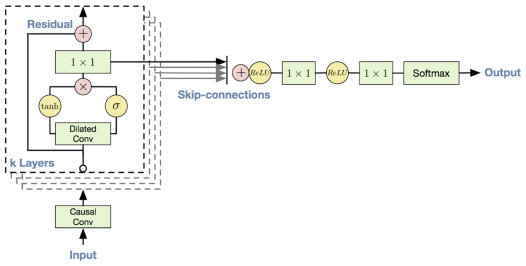

Figure 6: Residual Block and WaveNet Architecture

The implementation works as follows:

- **Input Preparation**: Usually, time series data is preprocessed before being fed into WaveNet. This could entail normalisation, scaling, and possibly transformations such as -law companding for audio data.
- **Stacking Dilated Causal Convolutions**: A sequence of dilated causal convolutional layers receive the preprocessed input. By extracting features at various time scales, each layer builds a representation of the temporal dependencies.
- **Passing through Gated Activation Units**: Each convolutional layer is followed by GLUs processing the output. Through selective information gating, the gate enables the model to suppress noise and concentrate on pertinent features.
- **Residual Connections**: Using residual connections, the output of each layer is added to the matching input. By doing this, the network's information flow is enhanced and the vanishing gradient problem is partially resolved.
- **Prediction and Loss Function**: The last layer generates a forecast for the subsequent time interval. The selected loss function (such as Mean Squared Error) between the expected and actual values is minimised during the training process.

In summary, WaveNet's distinct architecture and operation, along with the adaptability and features provided by GluonTS, combine to make it an effective tool for time series forecasting. WaveNet is capable of capturing long-range dependencies and learning complex patterns within the data by utilising dilated causal convolutions, GLUs, and residual connections. This allows for forecasts that are reliable and accurate.

These five models are well-suited for our comparative analyses since they are derived from different sources (such as audio analysis, bespoke time series models, etc.) and employ different techniques. We have detailed our methodology in great detail for two of these methods—Autoformer and DLinear—while leaving out the other three because their formats were similar to DLinear's. See Appendix 1 for a more thorough examination of our implementation and some sample code.

Our experiment was created to assess how well these models performed when examining a collection of solar energy data from the GluonTS library, which is accessible to the general public. One simple example of a time series dataset is solar energy levels. Forecasting solar energy levels would be helpful for many energy and climate-related applications.

4 Results

The results of our testing were not all the same. The Mean Absolute Scaled Error (MASE), which displays the relative error rates of each model, was used to calculate the forecasting error of each model. Although some of these MASE values are relatively high, forecasting is a challenging task overall, so these results are nevertheless a promising beginning.

We used the 'plot_gluonts' function in our research paper to visualise the predicted and actual time series data. Plots with shaded areas indicating prediction intervals are produced by this function and are essential for probabilistic forecasting. Two

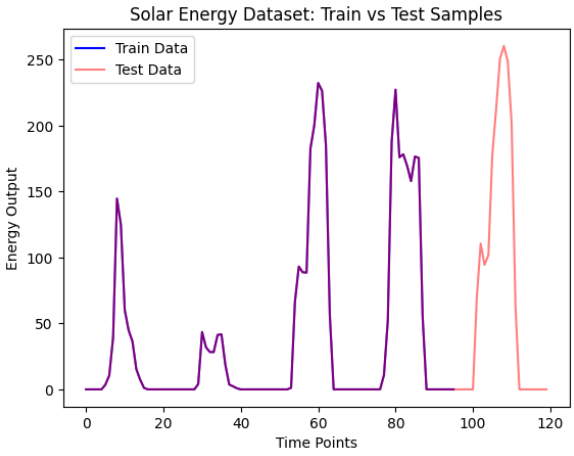

Figure 7: solar-energy data

different shades of green represent these intervals, which include the 50% and 90% prediction intervals. The model's prediction for the middle 50% of future values is represented by the 50% interval (narrower shade), which indicates a median level of uncertainty. Greater uncertainty is accounted for by the 90% interval (wider shade), which spans a wider range and shows where 90% of future observations are likely to occur. The understanding of the model's confidence and prediction variability is aided by this color-coded visualisation.

4.1 Autoformer

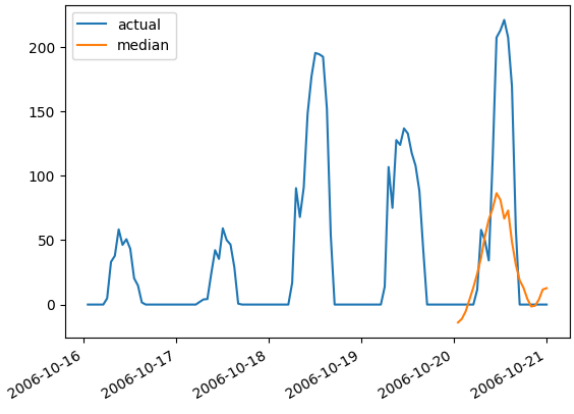

Figure 8: Autoformer Results

The transformer model's autoformer variation has had a big impact on NLP research. The transformer-based models' potential applicability in time series forecasting is indicated by their encouraging MASE value of 1.650.

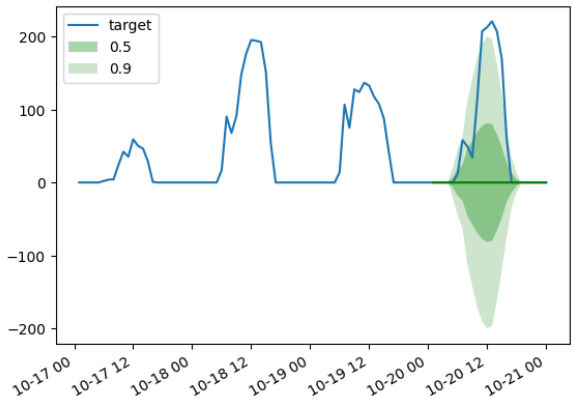

Figure 9: DLinear Results

4.2 DLinear

DLinear (MASE: 2.350): Like the SimpleFeedForward, DLinear's performance raises the possibility that more complex models—like RNNs or CNNs—might be required to fully capture the complexities of time series data.

4.3 DeepAR

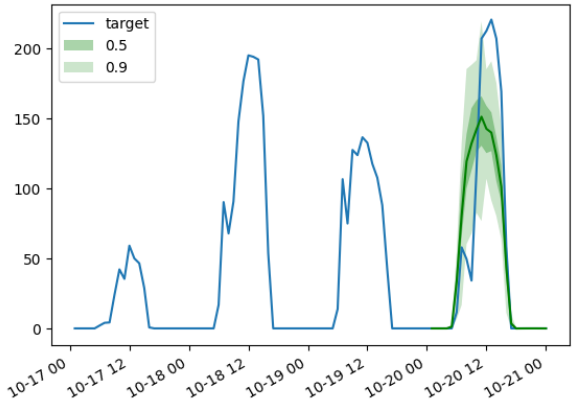

Figure 10: DeepAR Results

DeepAR (MASE: 1.262): Demonstrates adequate forecasting accuracy, highlighting the flexibility of recurrent neural network architectures—a popular approach in natural language processing—for time series data.

4.4 SimpleFeedForward

SimpleFeedForward (MASE: 2.351): The model's poorer performance suggests that more complicated time series forecasting tasks might be better suited for more straightforward neural network architectures.

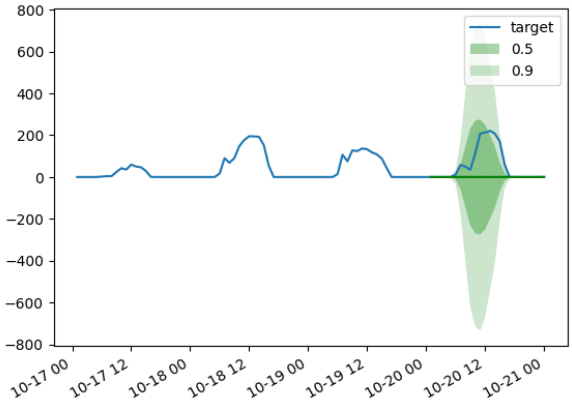

Figure 11: SimpleFeedForward Results

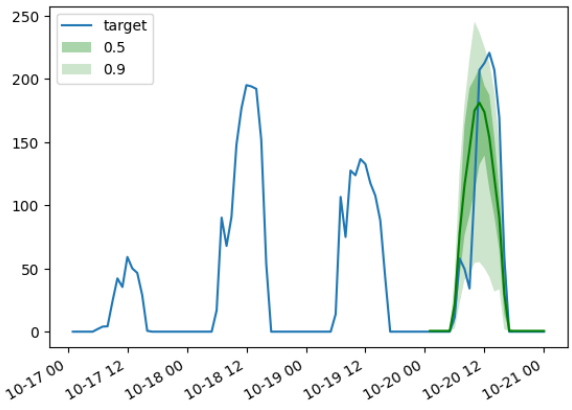

Figure 12: WaveNet Results

4.5 WaveNet

WaveNet (MASE: 1.163): With its maximum accuracy, this model demonstrates how well convolutional neural network methods—which are commonly used in deep learning and natural language processing—work for time series forecasting.

4.6 Model Comparison and MASE Scores

The Mean Absolute Scaled Error (MASE) scores of the various models offer important insights into their forecasting capabilities, as a summary of our findings. With a MASE of 1.650, the Autoformer shows the advantages of transformer-based models for representing temporal dynamics, but its higher score raises the possibility of constraints in our particular dataset. Because of their linear and basic architectures, DLinear and SimpleFeedForward have higher MASE scores (2.350 and 2.351, respectively), indicating difficulties in modelling nonlinear patterns. By comparison, WaveNet performs exceptionally well, having the lowest MASE of 1.163, demonstrating the effectiveness of convolutional networks in identifying intricate temporal relationships. Though marginally less successful than WaveNet, DeepAR, with a MASE of 1.262, also exhibits promise, particularly in capturing sequential dependencies.

These findings—which are graphically depicted in our plots—direct the process of choosing suitable models for various forecasting scenarios, highlighting the necessity of matching model capabilities to dataset complexity.

5 Future Work

Our project constitutes a proof of concept for the idea of applying NLP technology to time-series forecasting. The promising results suggest many avenues for further research. In particular, we have theorized a potential Sparse-Dense Attention Mechanism, which would aim to optimize the balance between the computational efficiency of sparse attention and the temporal precision of dense attention mechanisms. The core idea is to develop a Transformer that dynamically alternates between sparse and dense attention mechanisms as needed, utilizing the advantages of both while minimizing their drawbacks.

6 Conclusion

This project showcases the efficacy of NLP techniques and deep learning frameworks in the field of time-series forecasting. We selected a variety of models, including DeepAR, SimpleFeedForward, WaveNet, DLinear, and Autoformer. Each embodies different principles of deep learning and NLP. We then tested them on the same solar energy dataset and compared the results. This comparative analysis demonstrates that deep learning frameworks, particularly those leveraging recurrent and convolutional structures, can be useful for time-series forecasting. Models developed for sequential data processing in NLP tasks can indeed be successfully applied to the challenge of time-series forecasting, as evidenced by the performance of WaveNet and the encouraging outcomes from Autoformer. Building on these findings, one can envision a wealth of interesting applications, from financial markets to weather prediction. We also see pathways for further exploration and refinement, especially in advanced models like transformers, to fully exploit NLP system capabilities in forecasting scenarios.

7 Ethics

Time series forecasting is potentially a very powerful tool and has many potential ethical implications. We take this very seriously and have written this statement to draw attention for potential problem points. First, as with any system that makes use of large datasets, there is serious concern about privacy and security. Our testing was done using publicly available solar energy data, so we never handled any sensitive material, but future work must be careful about managing people's data. We are also concerned about potentials for bias in our models. Any model is going to contain and propagate biases found in its training data, and while we doubt our solar data contains such problems, applying our system to other datasets is likely to cause problems. Finally, this system is currently a proof of concept and should not be used for important decision making. The above problems mean that it is likely to make decisions that have unexpected problems that could lead to dire consequences. Further research and development is needed to ensure that all stakeholders' needs are respected.

8 Acknowledgments

This project was inspired by and builds upon the principles and methodologies developed in various seminal papers and articles in the field of time series forecasting and NLP. Special acknowledgment is given to the authors of these works for their contributions to the field. We would also like to thank our teacher, Professor Kenneth Church, for mentoring us throughout this project.

9 Individual Contributions

Research: mix of both, more Raman than Ed. The initial concept came from Ed, while much of the research related to specific models came from Raman.

Programming (see Appendix 1): Raman did this whole part, adapting and expanding upon concepts found in research. Whole coding and documentation on the Colab was also done by solely Raman.

Presentation: Raman took the lead here, doing much of the slide design with Ed doing review throughout. The presentation itself was split between them, with Raman taking the middle section (focusing on implementation and a technical explanation) while Ed did the start and end (intro, background, results, conclusion, and ethics).

Paper: Ed took the lead here, focusing on Introduction, Background, Results, Future Work, Conclusion, and Ethics, as well as the overall structure(Raman edited and reviewed later), while Raman did Methodology/Implementation (Ed did some structuring here too), and References. Both edited the whole paper together.

10 References

Simhayev, Eli, et al. "Yes, Transformers Are Effective for Time Series Forecasting (+ Autoformer)." *Hugging Face*, 16 June 2023, huggingface.co/blog/autoformer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

Gluonts - probabilistic time series modeling in python (no date) Static. Available at: https://ts.gluon.ai/stable/ (Accessed: 11 December 2023).

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J. and Sun, L., 2022. Transformers in time series: A survey. arXiv preprint arXiv:2202.07125.

Wu, H., Xu, J., Wang, J. and Long, M., 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in Neural Information Processing Systems, 34, pp.22419-22430.

Oord, A.V.D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. and Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.

Salinas, D., Flunkert, V., Gasthaus, J. and Januschowski, T., 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting, 36(3), pp.1181-1191.

Wu, H., et al., Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in Neural Information Processing Systems, 2021. 34.

Zhou, Haoyi, et al. "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting." Proceedings of the AAAI Conference on Artificial Intelligence, ojs.aaai.org/index.php/AAAI/article/view/17325. Accessed 10 Dec. 2023.

Deb, C., Zhang, F., Yang, J., Lee, S.E. and Shah,

K.W., 2017. A review on time series forecasting techniques for building energy consumption. Renewable and Sustainable Energy Reviews, 74, pp.902-924.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L. and Jin, R., 2022, June. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In International Conference on Machine Learning (pp. 27268-27286). PMLR.

Zhang, G.P. and Kline, D.M., 2007. Quarterly time-series forecasting with neural networks. IEEE transactions on neural networks, 18(6), pp.1800-1814.

Lim, B. et al. (2020) Temporal Fusion Transformers for interpretable multi-horizon time series forecasting, arXiv.org. Available at: https://arxiv.org/abs/1912.09363 (Accessed: 11 December 2023).

Preformer: Predictive Transformer with Multi-Scale Segment-Wise Correlations for Long-Term Time Series Forecasting. (2023, June 4). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/10096881

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A.X. and Dustdar, S., 2021, October. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In International conference on learning representations.

Cai, L., Janowicz, K., Mai, G., Yan, B. and Zhu, R., 2020. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. Transactions in GIS, 24(3), pp.736-755.

[Chen et al., 2022] Weiqi Chen, Wenwei Wang, Bingqing Peng, Qingsong Wen, Tian Zhou, and Liang Sun. Learning to rotate: Quaternion transformer for complicated periodical time series forecasting. In KDD, 2022.

[Chowdhury et al., 2022] Ranak Roy Chowdhury, Xiyuan Zhang, Jingbo Shang, Rajesh K Gupta, and Dezhi Hong. TARNet: Task- aware reconstruction for time-series transformer. In KDD, 2022.

[Liu et al., 2021] Minghao Liu, Shengqi Ren, Siyuan Ma, Jiahui Jiao, Yizhou Chen, Zhiguang Wang, and Wei Song. Gated trans- former networks for multivariate time series classification. arXiv preprint arXiv:2103.14438, 2021.

[Nie et al., 2023] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long- term forecasting with transformers. In ICLR, 2023.

[Shabani et al., 2023] Amin Shabani, Amir Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: iterative multi-scale refining trans- formers for time series forecasting. In ICLR, 2023.

[Wen et al., 2019] Qingsong Wen, Jingkun Gao, Xiaomin Song, Liang Sun, Huan Xu, et al. RobustSTL: A robust seasonal-trend decomposition algorithm for long time series. In AAAI, 2019.

[Wen et al., 2020] Qingsong Wen, Zhe Zhang, Yan Li, and Liang Sun. Fast RobustSTL: Efficient and robust seasonal-trend decom- position for time series with complex patterns. In KDD, 2020.

Ahmed, S., Nielsen, I.E., Tripathi, A., Siddiqui, S., Ramachandran, R.P. and Rasool, G., 2023. Transformers in time-series analysis: A tutorial. Circuits, Systems, and Signal Processing, 42(12), pp.7433-7466.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.X. and Yan, X., 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. Advances in neural information processing systems, 32.

Gardner Jr ES. 1985 Exponential smoothing: the state of the art. J. Forecast. 4, 1–28. (doi:10.1002/for.3980040103)

Ahmed NK, Atiya AF, Gayar NE, El-Shishiny H. 2010 An empirical comparison of machine learning models for time series forecasting. Econ. Rev. 29, 594–621. (doi:10.1080/07474938. 2010.481556)

Sen R, Yu HF, Dhillon I. 2019 Think globally, act locally: a deep neural network approach to high-dimensional time series forecasting. In Advances in Neural Information Processing Systems (NeurIPS), Vancouver, Canada, 8–14 December 2019.

Ghaderi A, Sanandaji BM, Ghaderi F. 2017 Deep forecast: deep learning-based spatio- temporal forecasting. In ICML Time Series Workshop, Sydney, Australia, 6–11 August 2017.

Young T, Hazarika D, Poria S, Cambria E. 2018 Recent trends in deep learning based natural language processing [Review Article]. IEEE Comput. Intell. Mag. 13, 55–75.(doi:10.1109/MCI.2018.2840738)

De Gooijer, J.G. and Hyndman, R.J., 2006. 25 years of time series forecasting. International journal of forecasting, 22(3), pp.443-473.

Chatfield, C., 2000. Time-series forecasting. CRC press.

Lim, B. and Zohren, S., 2021. Time-series fore-

casting with deep learning: a survey. Philosophical Transactions of the Royal Society A, 379(2194), p.20200209.

Torres, J.F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F. and Troncoso, A., 2021. Deep learning for time series forecasting: a survey. Big Data, 9(1), pp.3-21. Zhang, G.P., 2003. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, pp.159-175.

Bontempi, G., Ben Taieb, S. and Le Borgne, Y.A., 2013. Machine learning strategies for time series forecasting. Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures 2, pp.62-77. Wong, F.S., 1991. Time series forecasting using backpropagation neural networks. Neurocomputing, 2(4), pp.147-159. Taieb, S.B., Bontempi, G., Atiya, A.F. and Sorjamaa, A., 2012. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert systems with applications, 39(8), pp.7067-7083.

11 Appendix 1: Code Examples

Colab Link